KIT EDUCACIONAL

CURSO ARDUINO AUTOMATION

Manual do Usuário

Revisão 1 – Renato Aloi (07/10/2014)
Revisão 2 – Renato Aloi (19/11/2014)

2014 – Todos os direitos reservados

Sumário

Descrição do Kit Curso Arduino Automation....4
Introdução....4
Conteúdo do Kit Curso Arduino Automation....4
Central de Controle Automation....5
Painel de Controle Web....5
Conexão Com Rede/Internet....6
Novidades e Atualizações....7
Requisitos de Sistema....8
Termos Utilizados Neste Manual....9
Apresentação Dos Componentes....16
Tabela dos Códigos de Componentes....17
Caixa VDI Tigre....19
Placas / Módulos....20
Arduino Supervisório....21
Módulo Dimmer AC Bivolt....22
Distribuidor de Alimentação Supervisório....23
Adaptador Pino/Borne Supervisório....24
Módulo de Relê 4 Canais....25
Módulo Ethernet ENC28J60....26
Chicotes....27
Chicote Principal Automation....28
Chicote de Alimentação Automation....31
Chicote Fechadura 12V Automation....33
Rabichos....35
Rabichos de Alimentação....36
Rabichos de Comunicação....37
Rabichos de Conexão....38
Rabichos Especiais....40
Furação....41
Cortes....42
Acessórios....43
Fonte de Alimentação 12V / 1A....44
Expansores de Portas Automation....45
Montagem & Manutenção Geral....46
Material Necessário....46
Instalação da Caixa VDI....47
Furação e Corte da Caixa VDI....48
Montagem das Placas no Painel....49
Diagramas de Conexões dos Rabichos....51
Rabichos de Alimentação....52
Rabichos de Comunicação....54
Rabichos de Conexão....56
Rabichos Especiais....64
Diagramas de Conexões dos Chicotes....67
Chicote A – Chicote de Alimentação Automation....68
Chicote C – Chicote Principal Automation....70
Chicote D – Chicote da Fechadura 12V Automation....71

Instalação dos Programas & Drivers....73
Instalando Drivers do FTDi Basic no Windows....75
Instalando o Ambiente do Arduino no Windows....90
Instalando Drivers do FTDi Basic no MAC OS X....97
Instalando o Ambiente do Arduino no MAC OS X....108
Instalando Drivers do FTDi Basic no Linux....112
Instalando o Ambiente do Arduino no Linux....114
Configurando e Carregando o Painel de Controle Web....119
Instalando Bibliotecas Automation no Windows....120
Instalando Bibliotecas Automation no MAC OSX....139
Instalando Bibliotecas Automation no Linux....148
Configurando e Carregando o Código dos Dimmers....151
Instalando Código dos Dimmers AC....152
Utilização do Kit Automation....156
Configurações Básicas....157
Gerenciando Botões Automation....159
Gerenciando Dimmers Automation....161
Gerenciando Sensores Automation....165
Gerenciando Logins Automation....166
Gerenciando Banco de Dados WebDB Automation....168
Problemas Conhecidos....171
Ligações Na Rede Elétrica....172
Alimentação Principal....173
Diagramas de Ligações do Chicote Principal....174
Ligação da Iluminação – Forma Direta....176
Ligação da Iluminação – Com Interruptor Em Paralelo....177
Ligação da Iluminação – Direta Com Controle de Intensidade....178
Ligação da Iluminação – Com Controle De Intensidade E Interruptor Em Paralelo....179
Ligação da Iluminação – Abajur De Leitura....180
Ligação Da Ventilação – Controle Da Ativação, Velocidade e Sentido....181
Ligação Da Ventilação – Com Interruptor Em Paralelo....182
Sugestão de Passagem da Fiação....183
Anexos....184
Anexo 1 – Lista de Componentes....184
Lista Geral....184
Conteúdo da Caixa do Sistema VDI Tigre....184
Conteúdo da Caixa de Componentes Serial Link #1....185
Conteúdo da Caixa de Componentes Serial Link #2....185

Descrição do Kit Curso Arduino Automation

A ***Serial Link*** orgulhosamente apresenta o ***Kit Curso Arduino Automation***! Um kit educacional de componentes e placas para você automatizar sua casa! Experimente a sensação de controlar aparelhos da sua casa, ao toque de um dedo, na tela do seu *Celular* ou *Tablet*!

Compatível com todos os dispositivos *Android* ou *iOS*! Bastando apenas um Navegador de Internet para acessar a interface do ***Painel de Controle Web***.

Introdução

Kit de peças e componentes, contendo todo o material necessário para montar a ***Central de Controle Automation***, apresentada nas aulas do ***Curso Arduino Automation***.

Consiste em um conjunto de componentes, DVD's, uma caixa e um manual. Este manual será impresso, encadernado e enviado junto ao Kit, para que a pessoa, ao montar, possa manusear fisicamente o manual.

A ***Central de Controle Automation*** fornece as seguintes características:

- 3 dispositivos AC 110/220V (lâmpadas, ventiladores, geladeira, etc);
  - Sendo 2 dispositivos analógicos (lâmpadas e motores) com acionamento proporcional!
- 1 dispositivo DC de 12V (fechadura ou solenóides), até 1 ampère;

Com o ***Kit Curso Arduino Automation***, é possível efetuar o acionamento de uma lâmpada, um ventilador e/ou um portão eletrônico. A velocidade do ventilador pode ser controlada por um ***Dimmer AC Bivolt*** e, caso a lâmpada seja de filamento, podemos controlar seu brilho, utilizando o segundo ***Dimmer AC Bivolt***, que acompanha o Kit!

Atenção! Apenas lâmpadas de filamento podem ser ligadas nos ***Dimmer AC Bivolt,*** sendo que lâmpadas com reatores ou conhecidas por fluorescentes devem ser ligadas no ***Módulo de Relê de 4 Canais.***

Conteúdo do Kit Curso Arduino Automation

A lista de material contida no ***Kit Curso Arduino Automation*** segue abaixo:

- 1x Caixa VDI da Tigre
- 1x Kit Chicote Principal Automation (Saídas Dimmer, Ventilador e Iluminação)
- 1x Kit Chicote Alimentação Automation (2 Tomadas, com alimentação dos Dimmers)
- 2x Fontes de Alimentação 12V / 1A
- 1x Kit Chicote Fechadura 12V Automation
- 1x Kit Rabichos da Central de Controle Automation
- 2x Arduino Supervisório REV2 (Novo! Versão Red Dragon!)
- 1x Arduino FTDi Basic (Nova versão! Não precisa mais do rabicho!)
- 2x Distribuidor de Alimentação Supervisório
- 1x Adaptador Pino/Borne Supervisório
- 1x Módulo de Relê de 4 Canais
- 2x Módulo Dimmer AC Bivolt
- Manual do Usuario (este manual, impresso e encadernado)
- DVD's Curso Arduino Automation (4 DVD's)

Atenção! Esta lista serve apenas como referência, verifique o ANEXO 1 para lista completa!

Central de Controle Automation

A central de controle, ou "*centralina*" como também é conhecida, é uma caixa onde as placas de controle ficam localizadas fisicamente. E também centraliza todas as conexões!

Painel de Controle Web

Painel com botões (grandes) para acionamento (com os dedos) dos dispositivos da casa ou cômodo, conforme figura:

Veremos mais detalhes sobre o ***Painel de Controle Web*** nas seções apropriadas deste manual.

Novo! Agora com *Sistema de Logins Automation* e *Formulário de Login*!

Conexão Com Rede/Internet

A conexão entre o ***Kit Curso Arduino Automation*** e a *Rede Local*, ou apenas *Rede* de computadores é feita através de um cabo conhecido pelo seu conector *RJ45*. Este cabo também, as vezes, é chamado de "*Par-Trançado"*. ou simplesmente "*Cabo de Rede*".

Veremos mais adiante que existem dois tipos de cabos e que utilizamos:

- um tipo para ligar o *Kit Automation* no *Roteador*;
- mas também podemos utilizar o outro tipo para ligar diretamente em um computador, criando uma Rede apenas entre o *Kit Automation* e o *Computador*.

Para utilizar o *Kit Automation* na *Internet*, precisamos ligá-lo em um *Roteador* e configurar o *Encaminhamento de Portas*.

- Discuta e aprenda mais sobre *Encaminhamento de Portas* no Tópico do Fórum da Serial Link, segue o link: http://seriallink.com.br/forum/viewtopic.php?f=90&t=2018

Para ligar o *Kit Automation* em uma *Rede Local de Computadores*, precisamos de um *Cabo de Rede* normal e um *Roteador*. De um lado da conexão do *Cabo de Rede*, temos o ***Módulo Ethernet ENC28J60***; e na outra ponta, ligamos em um *Roteador*, ou diretamente em um computador.

Veremos mais detalhes sobre estas ligações, durante o manual. Vejamos uma foto do *Módulo Ethernet* (Rede Local), utilizado pelo ***Kit Curso Arduino Automation***.

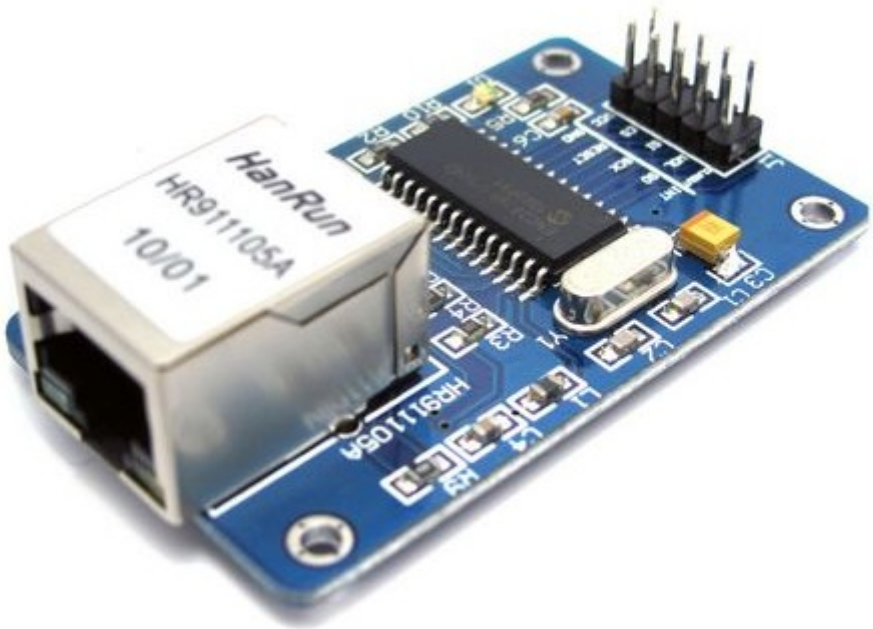


Foto de um ***Módulo Ethernet ENC28J60***, em destaque, o *Conector do Cabo de Rede*

Atenção! A conexão com a Internet é um assunto avançado e só será abordado no *Manual do Instalador*! Neste *Manual do Usuário*, nos limitaremos a detalhar o processo de conexão do ***Kit Curso Arduino Automation*** com a *Rede Local* de computadores; fazendo comentários sobre acesso a *Internet*, sempre que possível.

Novidades e Atualizações

Muitas novidades no lançamento do ***Kit Curso Arduino Automation***! A primeira e mais importante é o lançamento recente do ***Arduino Supervisório REV2***, com conector para o ***Arduino FTDi Basic***, eliminando assim o Rabicho FTDi que antes era necessário para carregar a programação no Arduino Supervisório!

Outra novidade importante é a segunda revisão do ***Chicote Principal Automation***! Totalmente remodelado, conta com fios diferenciados para facilitar as conexões, conforme veremos nesse manual!

Para se manter atualizado sobre as *Novidade*, *Atualizações* e *Lançamentos*, visite sempre o site:

http://www.seriallink.com.br

Fique por dentro das atualizações nos códigos e/ou bibliotecas, seguindo o endereço:

https://www.github.com/renatoaloi

Assista todos os vídeos do ***Curso Arduino Automation***:

- Nos DVD's que você recebeu, acompanhando o ***Kit Curso Arduino Automation***
- On-line no ***Canal do Curso Arduino*** no Youtube:
  https://www.youtube.com/user/cursoarduino

Playlist do ***Curso Arduino Automation:***

https://www.youtube.com/playlist?list=PLZBvA3l7rI6Fwbi5JSjnD-ONgS0JeQ5Rt

Participe do ***Fórum do Curso Arduino***:

http://seriallink.com.br/forum

Tópico no ***Fórum do Curso Arduino*** sobre Configuração de Roteadores:

http://seriallink.com.br/forum/viewtopic.php?f=90&t=2018

Tópico no ***Fórum do Curso Arduino*** sobre Personalizações da Biblioteca ***EthernetAutomation (novo!)***

http://seriallink.com.br/forum/viewtopic.php?f=50&t=2146

Atenção! Versão oficial da biblioteca *EthernetAutomation.h* (com login) já está no repositório, siga este link:
https://github.com/renatoaloi/EthernetAutomation

~~Na página 16 do tópico (link acima), você pode encontrar o download da versão não oficial da ***Biblioteca EthernetSup*** com sistema de *Login*! (ainda não lançado, no momento de produção desse manual).~~

Requisitos de Sistema

Para a instalação do ***Kit Curso Arduino Automation***, precisaremos de algumas ferramentas e também de um computador. Veja a lista de ferramentas na seção apropriada deste manual.

Vamos precisar de um computador para editar e personalizar o código do *Arduino*, conforme o gosto do cliente, alterando textos dos botões, ordem de aparição, funcionalidades adicionais, etc!

Podemos usar também o próprio computador para acessar o ***Painel de Controle Web***. Sim, existe a possibilidade de ligar a ***Central de Controle Automation*** diretamente no computador, sem a necessidade de um *Roteador*, mas discutiremos isso no *Fórum da Serial Link*.

Por enquanto precisamos apenas saber que o ***Arduino Supervisório***, que é o *Arduino* do ***Kit Curso Arduino Automation***, difere um pouco do ***Arduino UNO Italiano***, pelo fato de não possuir ***Conversor USB/Serial Integrado***. Por esse motivo, o *Kit Automation* é fornecido com o ***Arduino FTDi Basic***, que é uma ferramenta poderosa de integração *USB/Serial*.

A grande vantagem é que podemos utilizar apenas um único ***Arduino FTDi Basic*** para programar quantos ***Arduino's Supervisório*** quisermos!

Outras vantagens do ***Arduino FTDi Basic***:

- Não perde a programação, comum nos ***Arduino's UNO Italiano***, que acaba causando a perda do ***Conversor USB/Serial Integrado***, e consequentemente a perda da placa inteira. Isso não acontece com o ***Arduino FTDi Basic***!
- Por ser separado do ***Arduino Supervisório***, pode ser utilizado para outras tarefas, como interface de comunicação *Wireless*, *Bluetooth*, etc.
- Como o ***Arduino Supervisório*** não possui ***Conversor USB/Serial Integrado***, seu preço é bem mais em conta que um ***Arduino UNO Italiano***! Na verdade o preço de um ***Arduino Supervisório*** pode ser comparado a um ***Arduino UNO Chinês***!

Lista de Sistemas Operacionais Suportados pelo ***Kit Curso Arduino Automation*** (em breve Android!):

- **Windows XP/Vista/7/8** – O ***Arduino FTDi Basic***, produto utilizado para carregar os programas no ***Arduino Supervisório REV2***, é compatível com todas as versões do Windows, inclusive 64 bits!
- **MAC OSX 10.6+** – Pacote de instalação do *Driver FTDi* incluído no pacote de instalação do Arduino, muito fácil de instalar e usar (detalhes incluídos neste manual, adiante).
- **Linux Ubuntu 12.2 ~ 14.4** – Não precisa de nenhum driver adicional.

Atenção – Para acessar o ***Painel de Controle Web***, qualquer *Navegador de Internet* pode ser utilizado, variando apenas a aparência de bordas e definição do texto dos botões. Ou seja, apenas alterações nas características visuais.

Termos Utilizados Neste Manual

Durante o manual, utilizaremos vários termos. Ainda falta organizar essa seção por ordem alfabética, mas serve como um glossário e será revisado em futuras versões deste documento.

Central de Controle Automation

Termo que define a parte física do ***Kit Curso Arduino Automation***, compondo a caixa com seus componentes e fiação.

Arduino Supervisório

Arduino fabricado pela Serial Link para ser instalado em caixas e painéis. Vide Arduino, mais abaixo para maiores detalhes sobre o termo.

Este termo pode vir acompanhado de números de modelos e revisões, como é o caso do ***Arduino Supervisório REV2***, indicando a Revisão 2 da placa.

Painel de Controle Web

Termo utilizado para definir a página (Web) que utilizamos como painel de controle da nossa automação. Primeiro ajustamos e carregamos o código do Arduino (estude os termos!), depois acessamos o ***Painel de Controle Web*** para manipular nossos dispositivos, ligando lâmpadas e ventiladores.

Arduino FTDi Basic

Conversor USB/Serial que utilizamos para carregar nossos códigos no Arduino Supervisório. Somente através de um conversor desses que podemos comunicar o computador com o Arduino.

Este tipo de conversor necessita normalmente de um Driver, que precisa ser instalado no computador. Veja a seção de instalação de programas para maiores detalhes.

Driver

Programa que permite o computador “enxergar” algum dispositivo conectado nas portas USB, no nosso caso, o ***Arduino FTDi Basic***.

Dimmer AC Bivolt

Chamamos também de ***Módulo Dimmer AC Bivolt***, que é o módulo responsável por acionar cargas proporcionalmente. Através desse dispositivo, podemos controlar a intensidade de lâmpadas incandescentes, ou a velocidade de motores AC!

Módulo de Relê de 4 Canais

Aprendemos no ***Canal do Curso Arduino Básico*** que um Relê é um componente eletrônico, capaz de acionar cargas que não são compatíveis com o Arduino, ou seja, que queimariam nossos Arduinos!

Esses Relês precisam de circuitos de proteção para serem acionados, portanto tudo isso já está resolvido em um módulo que traz 4 Relês, com seus devidos circuitos de proteção, além de bornes parafusáveis que facilitam as conexões!

Bornes Parafusáveis

Alojamentos para conexões de fios, onde um parafuso mantém o fio firmemente conectado ao terminal da placa. Bornes normalmente são azuis e podem ser enfileirados formando vários alojamentos em sequência. Os alojamentos são todos identificados na impressão da placa. É o padrão adotado na plataforma Supervisório.

Conectores MODU

Termo utilizado para designar conectores fêmea (tipo alojamento) do padrão MODU, que é largamente utilizado na área de eletrônica em geral. É o padrão adotado na plataforma Arduino. São conectados em fios, formando Jumpers Fêmea, ou conectados em placas, formando barras de conectores fêmea.

Jumpers / Pinos Macho

Pinos macho são o oposto dos Conectores MODU, ao invés de alojamento, possuem pinos. Este termo está associado com Jumper que é um fio com terminais nas pontas. Esses terminais podem ser fêmea (MODU) ou machos (Pinos).

Roteador

Equipamento responsável por conectar computadores em uma rede. Além disso gerencia as conexões e efetua tarefas administrativas da rede. Os modelos atuais possuem poucas portas físicas, onde ligamos Cabos de Rede, mas aceitam muitos dispositivos Wireless conectados.

Wireless

Definição de rede sem fio, por rádio frequência. Roteadores utilizam essa tecnologia para conectar os computadores na Rede sem a necessidade de fios.

Ethernet / Internet / Rede

Estes são os termos que mais confundem os iniciantes! Utilizamos muitos termos quando nos referimos a redes de computadores. Quando estamos falando de uma Rede Local, ou seja, computadores ligados através de um Roteador, chamamos de Ethernet, ou simplesmente Rede Local, ou ainda só Rede.

Quando este Roteador, além de conectar os computadores em uma Rede Local, também está conectado a um Modem, dizemos que ele está conectado na Internet, portanto essa Rede possui acesso a Internet.

De uma forma geral, Rede é uma rede de computadores, Ethernet significa que essa rede é local, e Internet significa que ela é global.

Modem

Dispositivo normalmente fornecido pela operadora de serviços de Internet. No nosso caso do Brasil, pode ser a Telefônica, Net, Telemar, Claro, Vivo, etc.

Módulo de Ethernet ENC28J60

Módulo de Rede Local para o Arduino. Como já vimos, Ethernet significa rede de computadores no âmbito local, então podemos conectar nosso Arduino com outros computadores, através do Módulo Ethernet Enc28J60.

Arduino

O termo Arduino pode significar muitas coisas. Quando falamos Arduino em uma frase, sem mais nada acompanhando, normalmente estamos falando da plataforma, ou seja do conjunto da obra.

Quando falamos Arduino UNO ou Arduino Supervisório, estamos falando de um modelo específico de placa. Normalmente também esse tipo de termo com acessórios indica o fabricante, pois sabemos que Supervisório é uma marca da Serial Link, então Arduino Supervisório é o Arduino da Serial Link.

Quando dizemos *Programa do Arduino* ou *Arduino IDE*, estamos nos referindo ao programa de computador que utilizamos para programar nossos códigos do Arduino, e também para carregar nossos códigos no Arduino.

Já *Código do Arduino* significa o código que escrevemos e que será carregado no chip do Arduino. Esse processo de "carregamento" do código é explicado em outro termo. Esse código também pode ser chamado as vezes de *Sketch*.

Distribuidor de Alimentação Supervisório

Placa da Plataforma Supervisório que distribui a alimentação DC entre as outras placas do Kit. Possui um conector J4 para ligar a ***Fonte de Alimentação 12V / 1A*** e 5 saídas de bornes parafusáveis.

AC / DC

AC significa corrente alternada, que é o tipo de energia fornecida nas tomadas das nossas casas. Indica que não existe um positivo e um negativo fixos, eles alternam com o tempo, aqui no Brasil, em uma taxa de alternância de 60 ciclos por segundo (60Hz).

DC é a corrente contínua, ou direta. Nesse tipo de energia, temos um positivo e um negativo que nunca mudam. E normalmente são identificados pelas cores vermelho para o positivo e preto para o negativo.

Plataforma Supervisório

Placas produzidas pela Serial Link, utilizadas como módulos no ***Kit Curso Arduino Automation***. Cada placa possui uma funcionalidade específica. Verifique os termos para conhecer as funções de cada módulo.

Adaptador Pino/Borne Supervisório

Um módulo da *Plataforma Supervisório* responsável por converter as conexões do padrão MODU para o padrão de *Bornes Parafusáveis*.

Programa / Código

O termo "Programa" é bem confuso, pois utilizamos ele por várias vezes com sentidos diferentes. As vezes, significa um programa de computador, as vezes, um código criado por alguém. As vezes, ainda, significa um dispositivo que permite dois outros programas funcionar.

É um termo complicado até de explicar. Vamos usar exemplos. Na frase:

"Apareceu um erro na hora que fui carregar meu programa no Arduino."

Nesse exemplo, o sujeito está se referindo ao código que ele criou para Arduino.

"O programa travou e não responde mais!"

Nesta outra frase, vemos uma utilização totalmente diferente do termo “Programa”, que continua sendo um código desenvolvido por alguém, mas nesse caso, o sujeito da frase está reclamando que o programa que ele estava utilizando, travou. Podemos perceber que ele está reclamando do que chamamos de “Programa de Computador”, ou ainda “Software”.

Nesta segunda frase, o sujeito pode até estar se referindo ao *Arduino IDE* que pode ter parado de responder. Justamente nesse caso onde temos as maiores confusões.

Vamos a um terceiro exemplo:

“Apareceu um erro novo no programa do Arduino, quando fui carregar o programa Blink.”

Esta frase normalmente não será utilizada neste documento, mas apenas como exemplo, temos um usuário reclamando que a IDE do Arduino travou quando ele foi carregar o código (as vezes chamamos de Sketch) do Blink (código exemplo que vem junto com a IDE do Arduino). Repare como é complicado explicar! Vamos evitar esse tipo de confusão neste manual!

Arduino IDE

IDE é um termo largamente utilizado no mundo da programação, mas não existe tradução direta para o português. Mas é um termo em inglês que significa “*Interface Integrada de Desenvolvimento*”. Este termo define um ambiente onde podemos digitar nossos códigos; mas não apenas isso, também possuímos ferramentas de compilação, carregamento de códigos e monitores de depuração.

Carregar / Subir / Upload

Estes termos utilizamos quando estamos nos referindo a publicar nosso código, pronto, no chip do Arduino. Neste caso, quando digo Arduino, pode ser um Arduino UNO, Leonardo, Supervisório, não importa. A palavra que importa nessa frase é chip! Veja o termo para entender melhor.

Chip / Microcontrolador

Quando utilizamos estes termos, estamos nos referindo ao circuito integrado principal da placa. Então quando falamos chip do Arduino, estamos nos referindo ao integrado que faz o papel de cérebro do sistema. Então, em um Arduino UNO, o chip é o ATMega328p, no caso do Arduino Supervisório, também (vide a documentação para ver tabela comparativa), já no caso do Arduino Leonardo, por exemplo, o chip principal é o ATMega32u4.

Microcontrolador é um termo mais amplo, utilizado quando nos referimos a outras plataformas, por exemplo, na frase abaixo:

“O Arduino vem se popularizando no mercado de microcontroladores.”

Repare que o termo microcontrolador é utilizado para definir uma categoria de pequenos computadores baseados em chips.

Conversor USB / Serial

Termo utilizado para definir um adaptador físico, utilizado na conversão da comunicação entre o computador e um microcontrolador.

Os microcontroladores em geral, utilizam um tipo de comunicação Serial conhecida por TTL, mas os computadores utilizam USB. Um dispositivo que converte a comunicação USB para Serial TTL é conhecido por *Conversor USB/Serial*.

Serial TTL / Serial RS232 / USART

Termos utilizados para definir a comunicação serial entre dois equipamentos. Serial é uma palavra que existe em português e significa “em série”. É simples assim. As comunicações são consideradas em duas grandes categorias: *Paralela* e *Serial*.

Comunicação paralela trabalha em um modelo de envio das informações todas juntas ao mesmo tempo.

Comunicação serial utiliza o modelo de fila indiana, passando um por vez, em série!

A comunicação serial é a utilizada pelos microcontroladores e precisamos de um adaptador para tal função. Essa comunicação é chamada Serial TTL, pois é compatível com o padrão CMOS de construção de chips.

Já a Serial RS232 é um padrão antigo e caiu em desuso após a eliminação da porta DB9 dos computadores.

A Serial RS232 foi engolida pela USB e só através de um conversor podemos USB podemos nos comunicar com ela.

Ainda sim existem conversores RS232 como o MAX232, caso você esteja utilizando um computador antigo, vide este produto:

http://seriallink.com.br/loja/product_info.php?products_id=109

USB

Termo em inglês, acrônimo para “*Linha Universal de Comunicação Serial*”. Olha o termo Serial de novo aí! Sim USB é em série! E é um tipo de comunicação utilizada por computadores com Sistema Operacional, como Windows, Linux ou MAC OSX.

Cabo de Rede

Cabo responsável por interligar computadores e dispositivos em Rede. Utiliza conectores no padrão RJ45 e pode ser de dois tipos: normal ou cruzado (cross-over).

Cabos cruzados são poucos utilizados, apenas para ligar um computador em outro, criando uma Rede de apenas 2 computadores.

Cabos normais devem ser usados em redes com Roteadores. O Roteador é uma central de despachos de pacotes de comunicação e todos os computadores devem ser ligados nele através de Cabos de Rede para que estes computadores estejam na mesma Rede.

Só para conhecimento, antigamente, para ligar mais de dois computadores em Rede, utilizava-se cabos coaxiais e terminadores de 50 ohms. Quando uma conexão caia, a Rede inteira caía!

Cabo USB

Cabo de diversos tipos e formatos para ligar dispositivos compatíveis com esta tecnologia. O formato mais comum é conhecido por “boca de pato”, pois parece um pico de pato. Esse é o tipo A, temos também o tipo B, então é comum citarmos um Cabo USB tipo A / B Mini. Significa um cabo que tem em uma ponta um conector tipo A e na outra ponta um tipo B Mini. Existem os tipos Normal, Mini e Micro.

Energia Elétrica / Rede AC

Estes termos são usados quando nos referimos a energia elétrica que chega nas tomadas de nossas casas. É do tipo corrente alternada (AC) e pode matar! Normalmente opera em tensões entre 110V até 440V, dependendo das fases que chegam na residência, comércio ou indústria. Vamos focar na energia elétrica comumente usada no Brasil de 220V ou 110V em algumas regiões.

Tensão / Corrente / Resistência

Tensão é a grandeza medida em Volts. Corrente é o fluxo de energia, medida em Ampéres. Resistência é justamente a resistência que um corpo oferece a passagem de corrente.

Estes termos estão intimamente ligados pela lei de Ohm:

```
Tensão = Corrente x Resistência
```

Ou seja, Tensão é igual ao produto da Resistência pela Corrente, normalmente expresso por:

```
E = R . i
```

Onde `E` é a Tensão, `R` a Resistência, e `i` a Corrente.

Chicotes

Neste manual utilizamos o termo *Chicote* para definir um conjunto de fios, cabos e conectores, com um propósito em comum.

Por exemplo, o *Chicote de Alimentação Automation* é um conjunto de fios e tomadas que garantem alimentação ao circuito.

São diferentes dos *Rabichos*, pois oferecem alguma funcionalidade.

Rabichos

Este termo *Rabicho* vem sido utilizado nesse manual para indicar um conjunto de fios para ligar uma placa em outra, sem funcionalidades adicionais.

Outra forma que este termo pode ser usado é para se referenciar um pedaço de um *Chicote*. Por exemplo, o *Chicote de Alimentação Automation* é separado em 3 partes, que as vezes são chamadas de *Rabicho A*, *Rabicho B* e *Rabicho C*. Estas formas são utilizadas em textos mais antigos.

Conexões

Uma conexão normalmente envolve ligar duas ou mais pontas de um *Rabicho* ou de um *Chicote*. Por exemplo, quando falamos em conectar um *Rabicho*, precisamos na verdade ligar vários conectores de um lado e do outro da conexão.

Por isso, muitas vezes, simplificamos chamando de *conexão*, uma série de ligações que precisam ser feitas, seguindo uma ordem numérica ou de cores.

Barra Sindal

Barra de conectores parafusáveis para fios. Os *Bornes* são conectores parafusáveis para as placas e as *Barras Sindal* ou *Conectores Sindal*, como são chamados também, fazem a mesma função, só que para fios.

Apresentação Dos Componentes

Neste tópico verificaremos todos os componentes que fazem parte do Kit, aprendendo um pouco sobre cada um e, de quebra, conferindo as peças!

Tabela dos Códigos de Componentes

Esta tabela abaixo lista todos os Códigos utilizados nesse manual e cria alguns novos, melhorando assim a comunicação de Instaladores e Clientes.

Assim quando formos identificar uma peça ou localização de furação, podemos usar a tabela abaixo como referência de códigos do ***Kit Curso Arduino Automation***.

Por exemplo, existem frases como esta pelo manual: “Parafusar a Tomada T1 nos Furos FA1 e FA2, utilizando dos Parafusos PA1 e PA2”. Essa tabela abaixo serve para “traduzir” esse tipo de frase.

Qtde.	Código	Descrição	Obs.
2	SUPARDREV2	Arduino Supervisório REV2 (Red Dragon) Com conector FTDi!	Fabricação Serial Link
1	ARDFTDI	Arduino FTDi Basic	Fabricação Serial Link
1	SUPADA	Adaptador de Pinos/Bornes Supervisório	Fabricação Serial Link
2	SUPDIST	Distribuidor de Alimentação Supervisório	Fabricação Serial Link
1	MODREL4	Módulo de Relê 4 Canais	Fabricação Serial Link
2	MODDIMAC	Módulo Dimmer AC Bivolt	Fabricação Serial Link
1	ENC28J60	Módulo Ethernet Enc28J60	
1	Chicote A	Chicote de Alimentação Automation	Nova versão! Inclui antigo Chicote E
~~0~~	~~Chicote B~~	~~(ANTIGO) Chicote FTDi~~	~~Não é mais utilizado~~
1	Chicote C	Chicote Principal Automation	Nova versão! Novo Conector!
1	Chicote D	Chicote Fechadura 12V Automation	Nova versão! Inclui 2 partes.
~~0~~	~~Chicote E~~	~~(ANTIGO) Chicote dos Dimmers AC~~	~~Incluído no Chicote A;~~
3	Rabicho A	Rabichos de Alimentação	Rabichos da alimentação DC 12V
1	Rabicho B	Rabichos de Comunicação	Rabichos de Comunicação Serial, I2C, etc.
4	Rabicho C	Rabichos de Conexão	Rabichos de conexão em geral.
1	Rabicho D	Rabichos Especiais	Rabicho especial dos Dimmers AC.
2	Furo FA1, FA2	Sugestão de Furação	
	Furo FB1, FB2	Sugestão de Furação	
	Furo FC1	Sugestão de Furação	
	Furo FD1	Sugestão de Furação	
	Furo FE1	Sugestão de Furação	
	Corte AC1	Sugestão de Corte	
2	Tomada T1, T2	Fazem parte do Chicote “A”	
2	Fonte 12V FT1, FT2	Fontes de Alimentação 12V DC / 1A	
1	Caixa VDI1	Caixa do Sistema VDI da Tigre	
1	Fechadura FH1	Fechadura Eletrônica	
4	Parafusos PA1, PA2,	Parafusos de fixação das Tomadas T1	

	PB1, PB2	e T2	
4	Parafusos PC1, PC2, PC3, PC4	Parafusos de fixação da Caixa VDI na parede	
4	Parafusos PD1, PD2, PD3, PD4	Parafusos de fixação da Tampa da Caixa VDI	
4	Buchas 8mm BA1, BA2, BA3, BA4	Buchas de fixação da Caixa VDI na parede	
25	Cintas Plásticas	Cintas plásticas com trava (braçadeiras)	

Caixa VDI Tigre

O ***Kit Curso Arduino Automation*** foi desenvolvido para ser montado dentro de uma *Caixa do Sistema VDI da Tigre* de 30x30 cm.

Veja maiores detalhes sobre a ***Caixa VDI 30x30 de Sobrepor na Parede***, utilizada no ***Curso Arduino Automation***, diretamente no site do fabricante:

http://www.tigre.com.br/

Esta caixa pode ou não ser fornecida juntamente com o ***Kit Curso Arduino Automation***, dependendo da versão adquirida. De qualquer forma, as versões do Kit que acompanham a Caixa VDI da Tigre, trazem a caixa intacta, conforme vendida pela *Tigre*, em seus revendedores.

Nenhuma furação é feita na caixa pela *Serial Link*, sendo que as furações citadas neste documento são apenas *sugestões*, conforme a montagem feita no curso. Cabe ao instalador decidir a melhor forma e as melhores posições dos furos, conforme a necessidade.

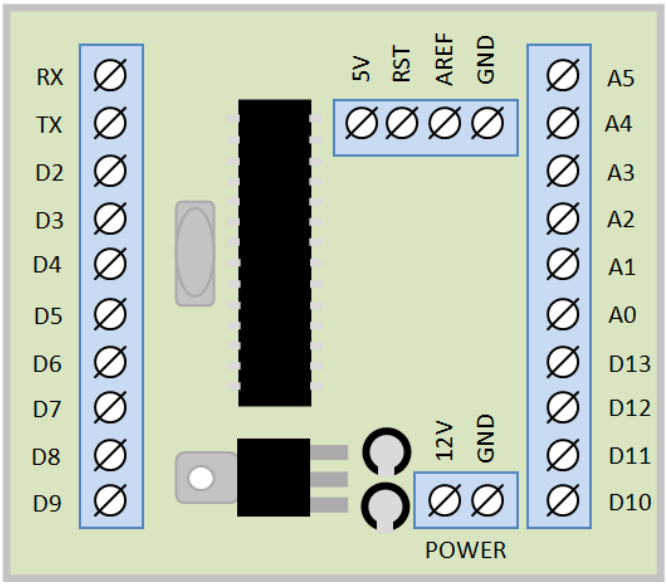
Placas / Módulos

As placas e/ou módulos fornecidos com o ***Kit Curso Arduino Automation*** seguem o padrão do Kit Básico, conforme segue:

- 2x Arduino Supervisório REV2 (ou superior)
- 2x Módulo Dimmer AC Bivolt
- 2x Distribuidor de Alimentação Supervisório
- 1x Adaptador Pino/Borne Supervisório
- 1x Módulo de Relê 4 Canais
- 1x Módulo Ethernet ENC28J60
- 1x Arduino FTDi Basic

Arduino Supervisório

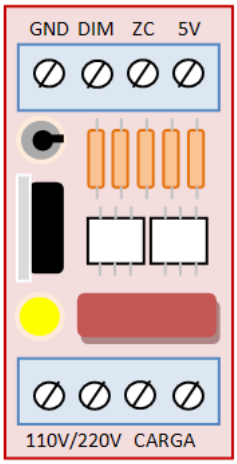
Independente da versão, a representação gráfica do Arduino Supervisório que veremos durante esse manual, é igual a figura abaixo:



A versão REV2, fornecida no momento que este texto foi escrito, possui um conector adicional no canto superior esquerdo, para ligação direta do ***Arduino FTDi Basic***. Veremos essa característica em momento oportuno. Continue lendo.

Módulo Dimmer AC Bivolt

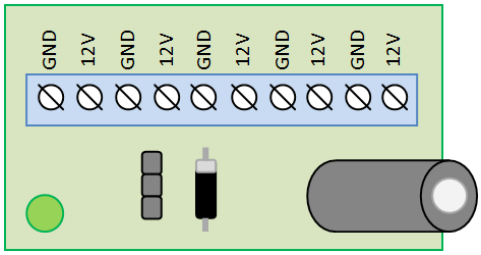
Os *Módulos Dimmer AC Bivolt* também já estão com nova versão que é um pouco diferente da representação gráfica desse documento, mas os conectores continuam no mesmo lugar. Vamos ver a imagem do Dimmer AC Bivolt abaixo:



Repare na versão que você tem em mãos, que alguns componentes mudaram de lugar para melhorar a resistência mecânica do módulo.

Distribuidor de Alimentação Supervisório

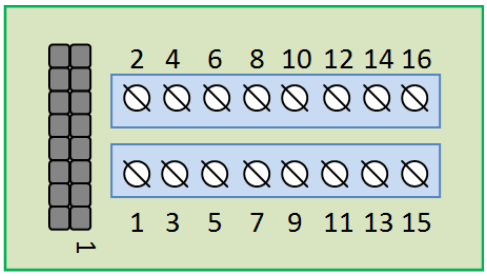
Módulo responsável por alimentar as outras placas com 12V DC. Possui uma entrada J4 para ligação da Fonte 12V / 1A e 5 saídas em bornes parafusáveis. Veja a representação gráfica na imagem abaixo:



Apesar das Saídas apresentarem marcações como 12V, depende da fonte utilizada. No caso do ***Kit Curso Arduino Automation***, a fonte é de 12V, então as saídas correspondem às marcações. Mas se uma fonte de 5V for utilizada, as saídas de bornes parafusáveis será também de 5V.

Adaptador Pino/Borne Supervisório

Este módulo apenas efetua conexões de tipos diferentes. Esta placa utiliza pinos no formato MODU Macho ligados diretamente a Bornes Parafusáveis, para integração de dispositivos no padrão Arduino com o padrão Supervisório.

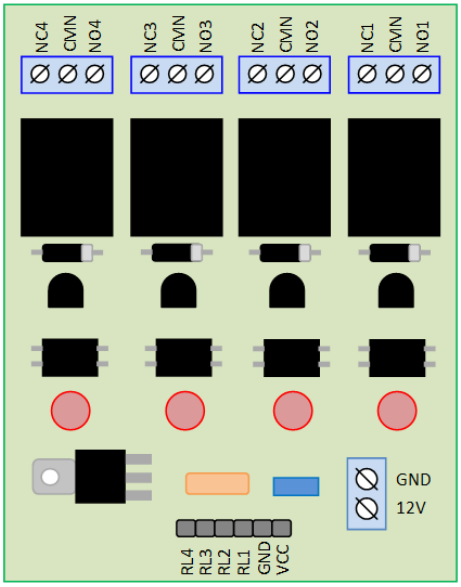


Módulo de Relê 4 Canais

Este módulo possui 4 saídas para acionamento de dispositivos que não são compatíveis com o Arduino, como lâmpadas, motores, solenóides etc.

Cada saída isola completamente tensões como 220V do Arduino, através de cicuitos de proteção com optoacopladores.

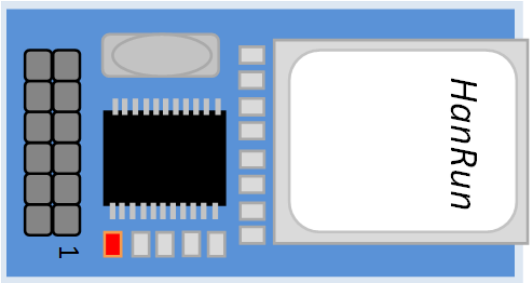
Vejamos uma representação gráfica do Módulo de Relê 4 Canais:



Repare no módulo que temos 4 saídas, cada com 3 terminais de bornes parafusáveis, identificados como `NC`, `CMN` e `NO`, representando respectivamente: ***Normalmente Fechado***, ***Comum*** e ***Normalmente Aberto***.

Módulo Ethernet ENC28J60

Módulo responsável por conectar o Arduino Supervisório na Rede Local. Devemos ligar esse módulo através de um Cabo de Rede em um Roteador.



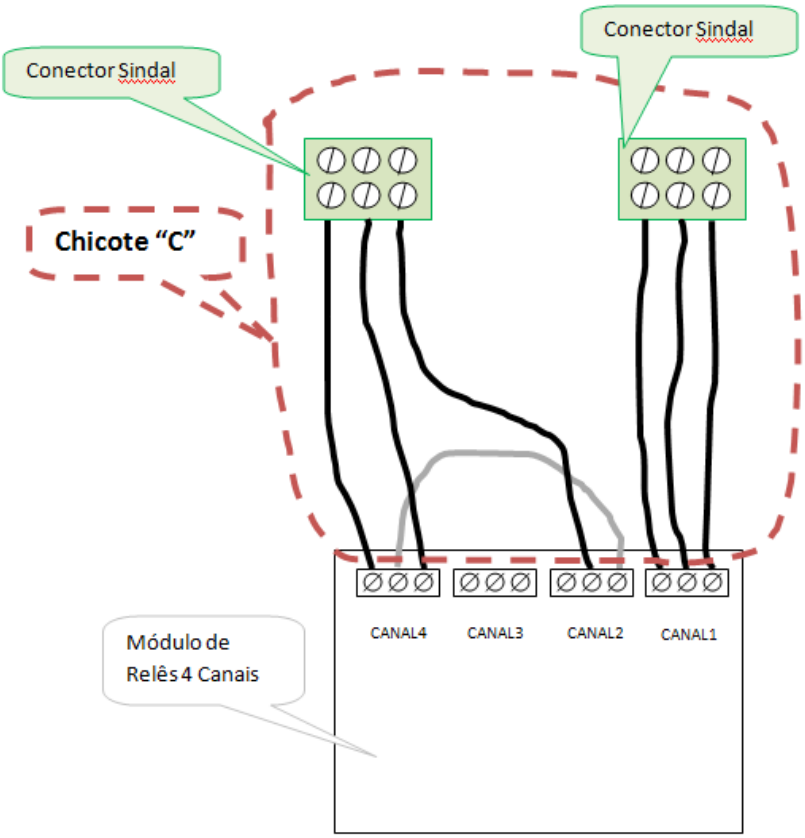
Este é o único módulo que não é fabricado pela Serial Link (por enquanto). Repare que seus pinos são ordenados com base no pino 1, marcado na imagem. Veremos mais sobre seus pinos na montagem.

Chicotes

Os chicotes são parte essencial do ***Kit Curso Arduino Automation***, pois eles garantem funcionalidades especiais. Por exemplo, o *Chicote de Alimentação*, fornece energia; já o *Chicote da Fechadura* auxilia na integração de solenóides com a plataforma *Automation*.

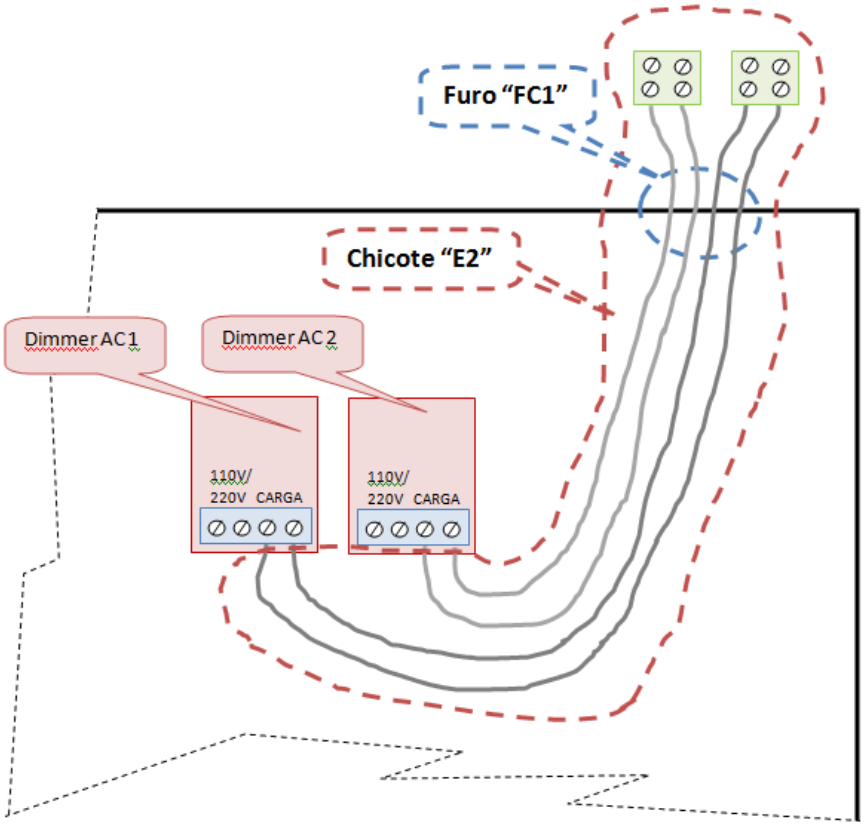
Chicote Principal Automation

Nova versão! Este chicote, conhecido por *Chicote "C"*, está sendo lançando juntamente com o ***Kit Curso Arduino Automation***, visando facilitar as ligações do ***Módulo de Relê 4 Canais*** e também dos ***Dimmer AC Bivolt***. Vejamos um esquema que ilustra onde este chicote é utilizado em relação às placas:

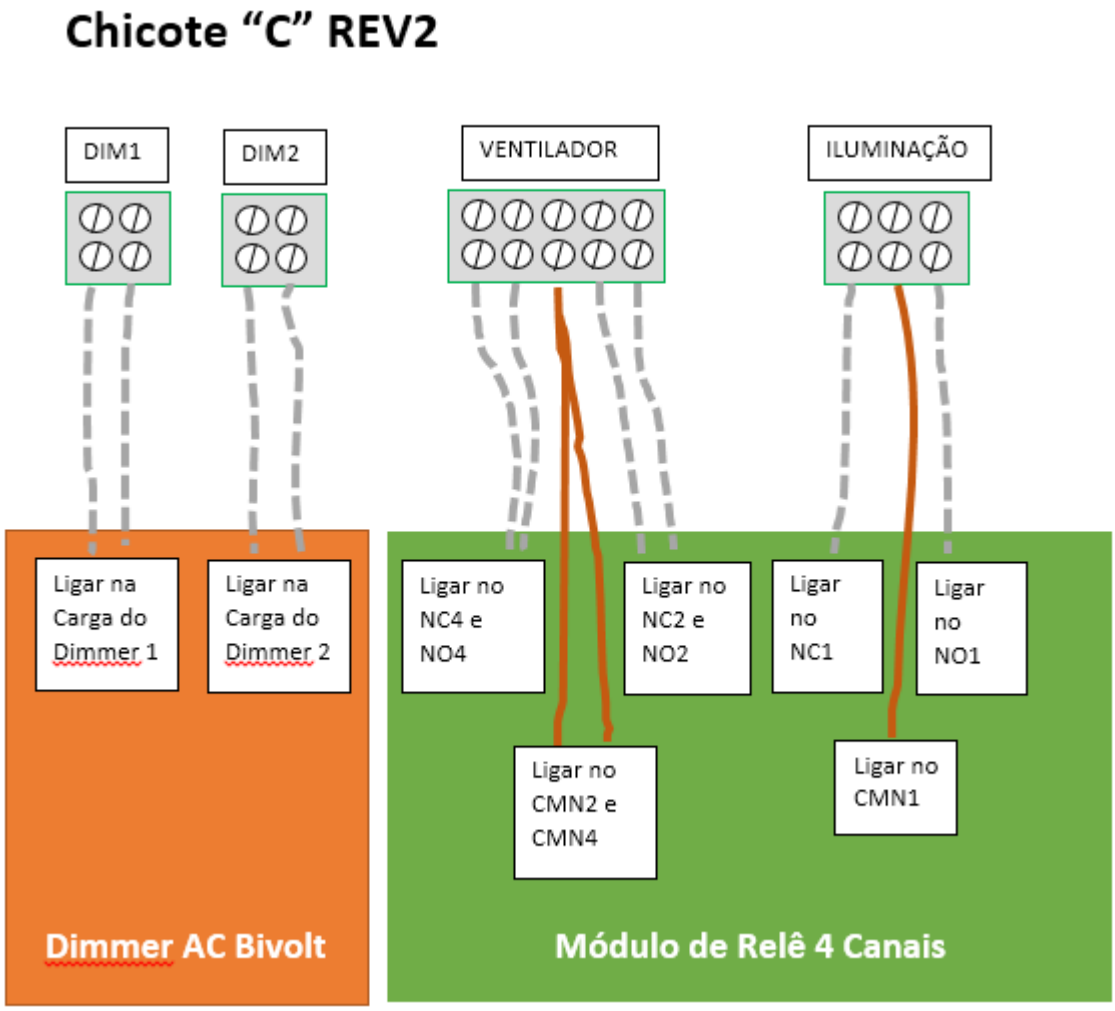


Este *Chicote "C"*, como é conhecido pelo seu código, é diferente do que é apresentado acima. Este é um diagrama antigo só pra entendermos onde o chicote vai conectado.

Atenção! Atualmente o *Chicote "E2"* (chicote das cargas dos dimmers) faz parte do chicote principal. Vejamos as conexões dos *Módulos Dimmer AC Bivolt*:



A versão atual do *Chicote "C"* tem o seguinte diagrama:



As linhas tracejadas, no diagrama, correspondem aos fios brancos do chicote; já as linhas marrons, correspondem aos fios de capa transparente, do chicote.

O ponto mais importante do diagrama é ligar os dois fios do lado esquerdo do conector Sindal principal, nos bornes NC4 e NO4; e os dois fios do lado direito, nos bornes NC2 e NO2.

Entre os fios da esquerda ou os dois da direita, não existe ordem certa para ligar, por exemplo, ao efetuar as conexões nos bornes NC2 e NO2, não importa qual dos dois fios branco da direita será ligado no NC2 e qual será ligado no NO2.

O diagrama considera o ponto de vista de "frente" do conector sindal, ou seja, olhado do lado com os parafusos.

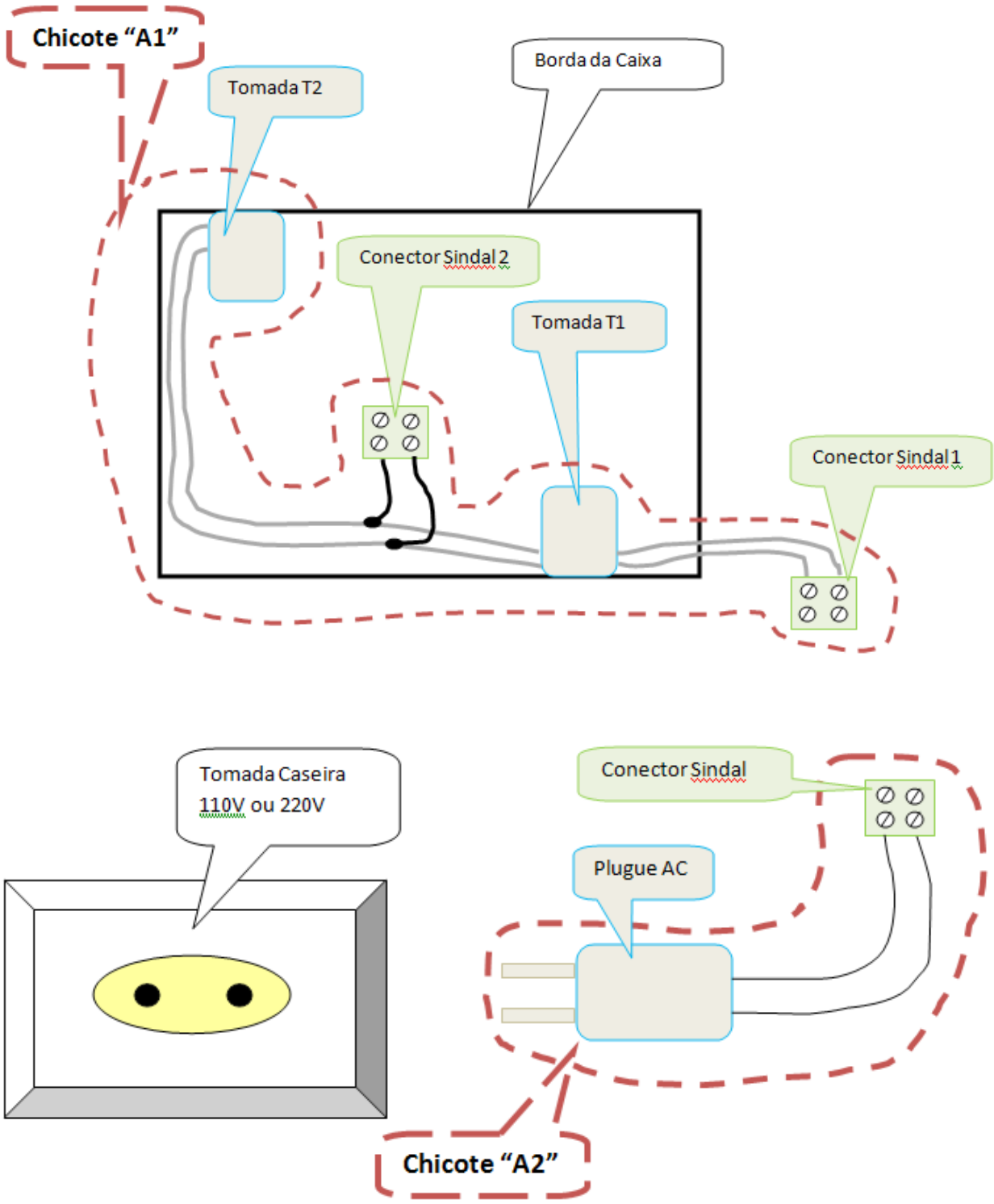
Uma das melhorias feitas no Chicote "C", foi a eliminação do Jumper, entre o CMN2 e o CMN4, substituído por um fio duplo, conectado no alojamento central do conector Sindal principal.

Outra melhoria foi o conector Sindal de 5 pinos, como sendo o principal. Agora o Chicote "C" já está pronto para ligação do interruptor em paralelo com o ventilador! Foi solicitado por muitos clientes!

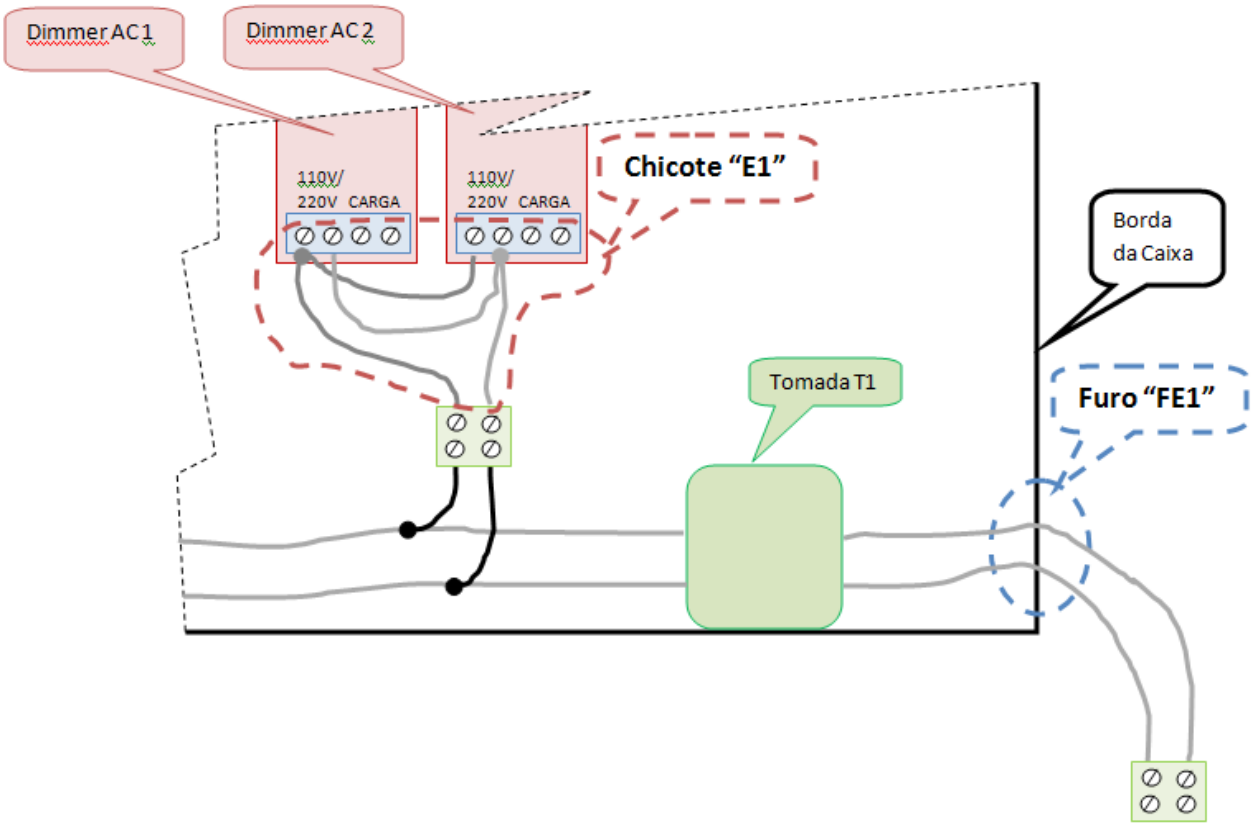
Produto acompanha o ***Kit Curso Arduino Automation***, mas também pode ser adquirido separadamente no link: http://seriallink.com.br/loja/product_info.php?cPath=61&products_id=349

Chicote de Alimentação Automation

Este chicote é responsável pela alimentação do ***Kit Curso Arduino Automation***. Ele é composto por 3 partes. Na documentação inicial este chicote era citado como *Chicote "A1"*, *Chicote "A2"* e *Chicote "E1"*. Vamos entender através dos esquemas de ligação. Vejamos primeiro o *Chicote "A"*, partes 1 e 2:



Agora vejamos o *Chicote "E1"*, que foi incorporado ao *Chicote "A"*:



- **Chicote "A"**
  - Formado por duas partes:
    - **Parte 1** – Composto por duas tomadas para fixação interna à caixa e rabicho para passar pelo *Furo "FE1"* (vide furação, mais abaixo); e
    - **Parte 2** – Plugue com rabicho, para ligar a caixa em uma tomada externa.
  - Veremos na parte da montagem que ligamos a Parte 1 do Chicote A na Parte 2, garantindo assim alimentação 110/220V para a ***Central de Controle Automation***.
- **Chicote "E1"**
  - Neste chicote ligamos a alimentação dos ***Módulos Dimmer AC Bivolt***.

O ***Chicote de Alimentação Automation***, portanto, é composto dessas 3 partes que vimos acima, nos diagramas.

Além de acompanhar o ***Kit Curso Arduino Automation***, este chicote também é vendido separadamente no seguinte link, em um kit que acompanha também as Fontes de Alimentação 12V / 1A:

http://seriallink.com.br/loja/product_info.php?cPath=61&products_id=352

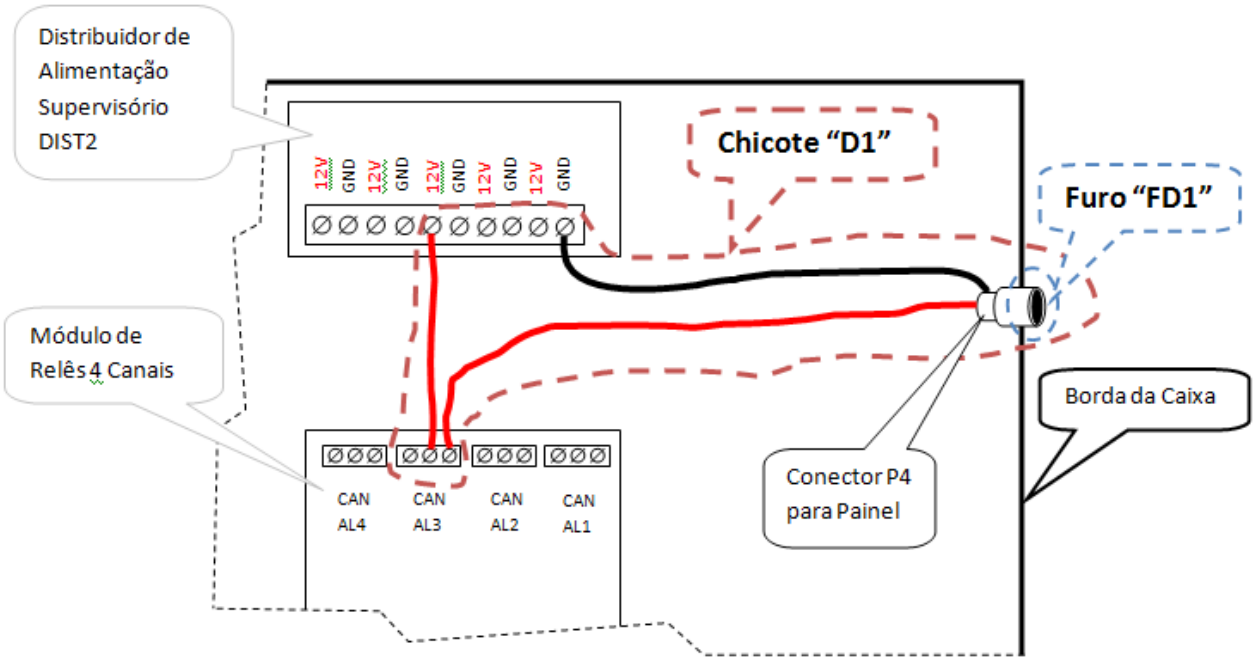
Chicote Fechadura 12V Automation

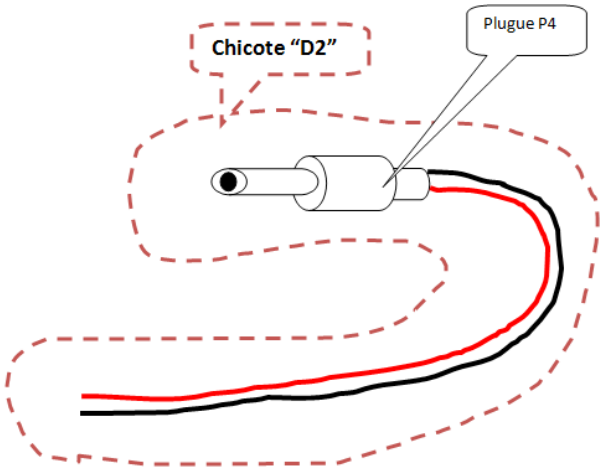
Este chicote é composto de duas partes: uma principal, e uma externa para ligação da fechadura. São 2 fios paralelos, um vermelho e um preto. Na ponta temos um plugue P4 para conexão com a Fechadura Eletrônica. O fio vermelho é 12V e o preto é Negativo, ou seja, 0V.

No plugue P4, temos positivo na parte central (interna) e negativo na parte externa. Veremos as formas de conexão na parte de montagem.

Vamos ver agora os diagramas desse chicote, também conhecido pela documentação antiga por:

- *Chicote "D"*:





Repare, nos diagramas acima, que esse chicote era separado em duas partes D1 e D2. Agora é somente chamado de ***Chicote Fechadura 12V Automation***, e compõe as duas partes.

Este chicote, além de fazer parte do ***Kit Curso Arduino Automation***, também é vendido separadamente no link:

http://seriallink.com.br/loja/product_info.php?cPath=61&products_id=351

Rabichos

Rabichos são um pouco mais complicados que chicotes, pois eles são menores e muito parecidos entre si! E para piorar não existem diagramas de conexão para eles, apenas tabelas de pinagem. Cabe ao instalador achar o fio certo, do rabicho certo, com a cor certa para conectar, conforme as tabelas que veremos.

Todos os rabichos necessários acompanham o ***Kit Curso Arduino Automation***, mas eles também podem ser comprados todos juntos, em conjunto, nesse link:

http://seriallink.com.br/loja/product_info.php?cPath=61&products_id=344

Os rabichos são separados pelo uso, existem de *Conexão*, de *Alimentação*, de *Comunicação*, e *Especiais*. Vejamos uma lista com todos os rabichos incluídos no ***Kit Curso Arduino Automation***:

- 3x Rabichos de Alimentação Supervisório
- 1x Rabicho de Comunicação Serial
- 4x Rabichos de Conexão Supervisório
  - 2x No Padrão MODU
  - 2x No Padrão Bornes (fio com ponta estanhada)
- 1x Rabicho Especial dos Dimmers AC Bivolt

Rabichos de Alimentação

São 3 no total e possuem apenas 2 fios, sendo um preto e um vermelho.

Cor	Descrição	Comprimento
Vermelho	Positivo (12V)	30 cm
Preto	Negativo (GND)	30 cm

Utilizados para ligar alimentação entre o Distribuidor de Alimentação Supervisório e as placas do Arduino Supervisório e do Módulo de Relê 4 Canais.

Rabichos de Comunicação

Temos apenas 1 no ***Kit Curso Arduino Automation***, que possui 2 fios, um verde e um preto. Esse rabicho é utilizado para ligar a Comunicação Serial entre o Arduino Supervisório que controla a Rede com o Arduino Supervisório que controla os Dimmers AC Bivolt.

Cor	Descrição	Comprimento
Verde	Comunicação Serial	30 cm
Preto	Negativo (GND)	30 cm

Rabichos de Conexão

Este tipo em especial de rabichos são mais complicados, pois possuem dois sub-tipos: com e sem terminais MODU. Ou seja, temos Rabichos de Conexão Machos, sem terminais, apenas com a ponta do fio estanhada; e do outro tipo, temos um terminal MODU fêmea em cada ponta.

Atenção! Por esse motivo acima, o ***Módulo de Relê 4 Canais*** do ***Kit Curso Arduino Automation*** é montado especialmente com pinos machos na placa. Caso você compre módulos avulsos, solicite os pinos machos para compatibilidade com a plataforma Automation!

Chamamos então os rabichos com as pontas soldadas de “Machos” e os rabichos com os terminais MODU de “Fêmea”. Pronto, esses são os dois sub-tipos, vamos adiante!

1x Rabicho Fêmea de 6 fios (RBF1)

Cor	Descrição	Comprimento
Azul	Negativo (GND)	20 cm
Branco	Cable Select (CS)	20 cm
Verde	Server Input (SI)	20 cm
Amarelo	Server Output (SO)	20 cm
Preto	Sincronia do SPI (SCK)	20 cm
Vermelho	Positivo (5V)	20 cm

1x Rabicho Fêmea de 5 fios (RBF2)

Cor	Descrição	Comprimento
Azul	Relê do Canal 1	20 cm
Preto	Negativo (GND)	20 cm
Branco	Relê do Canal 4	20 cm
Verde	Relê do Canal 3	20 cm
Amarelo	Relê do Canal 2	20 cm

1x Rabicho Macho de 6 fios (RBM2)

Cor	Descrição	Comprimento
Azul	Negativo (GND)	20 cm
Branco	Cable Select (CS)	20 cm
Verde	Server Input (SI)	20 cm
Vermelho	Positivo (5V)	20 cm
Preto	Sincronia do SPI (SCK)	20 cm
Amarelo	Server Output (SO)	20 cm

1x Rabicho Macho de 5 fios (RBM1)

Cor	Descrição	Comprimento
Preto	Negativo (GND)	20 cm
Azul	Relê do Canal 1	20 cm
Amarelo	Relê do Canal 2	20 cm
Verde	Relê do Canal 3	20 cm
Branco	Relê do Canal 4	20 cm

Rabichos Especiais

Temos apenas 1 desse tipo que é o ***Rabicho Especial do Dimmer AC Bivolt***, com derivação para ligação de 2 ***Módulo Dimmer AC Bivolt***.

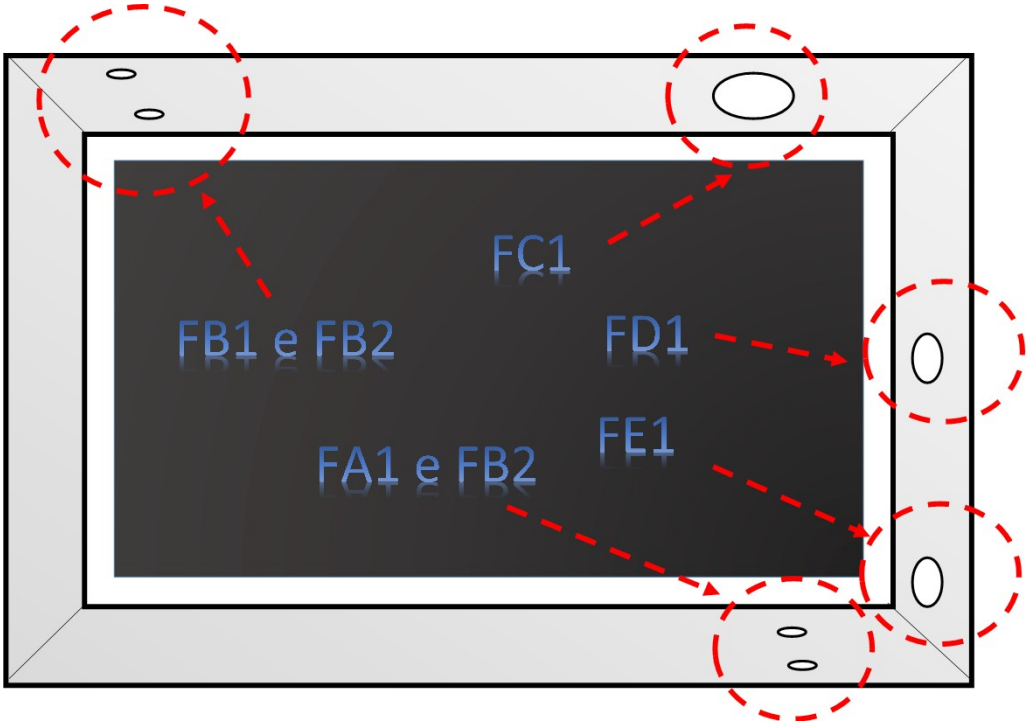
Cor	Descrição	Comprimento
2x Verde	Comunicação DIM1 e DIM2	30 cm
Preto	Negativo (GND)	30 cm
Vermelho	Positivo (5V)	30 cm
Branco	Aviso de Zero-Cross (ZC)	30 cm

Repare que existem dois fios Verdes, que são conectados nos *Bornes DIM* dos ***Módulo's Dimmer AC Bivolt***.

Furação

Conforme podemos ver no diagrama abaixo, temos códigos indicando cada sugestão de furo a ser feita pelo instalador, na hora da montagem.

Este diagrama de furações é apenas uma sugestão, baseada na montagem feita nas aulas do *Curso Arduino Automation*, no *Canal do Curso Arduino (Youtube: https://www.youtube.com/cursoarduino)*.



Vejamos abaixo a tabela de códigos e dimensões dos furos:

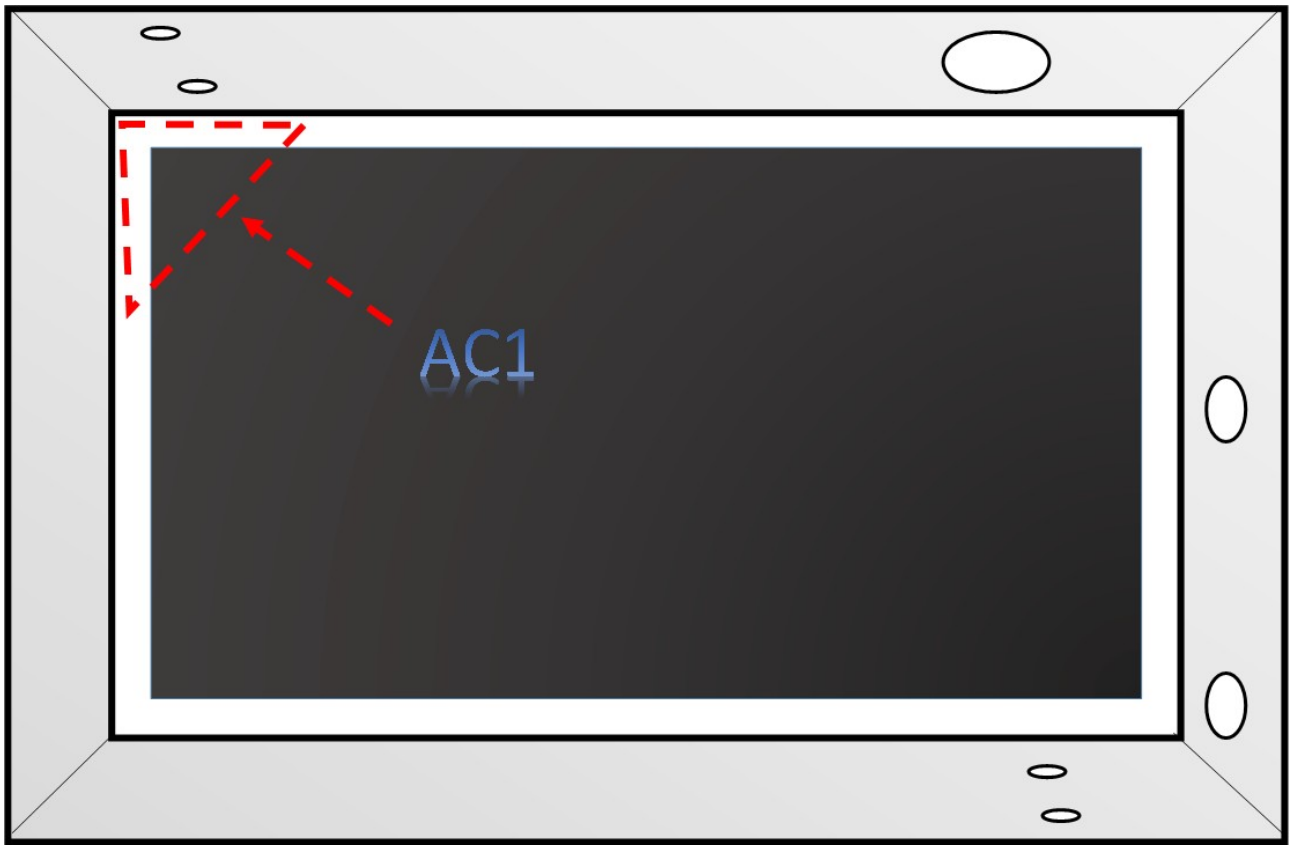
Código do Furo	Descrição	Diâmetro	Posição
FA1 e FA2	Furação de fixação da Tomada T1	2 ~ 3 mm	Utilize a própria tomada para marcar a posição dos furos.
FB1 e FB2	Furação de fixação da Tomada T2	2 ~ 3 mm	Utilize a própria tomada para marcar a posição dos furos.
FC1	Furo de passagem do Chicote Principal e do Cabo de Rede	25 mm	Painel superior, alinhado com a saída do Módulo de Relê.
FD1	Furo de fixação do Conector J4 de Painel (utilize a rosca para fixação)	12 mm	Painel lateral, mais ou menos no meio.
FE1	Furo de passagem do Chicote de Alimentação.	5 ~ 10 mm	Painel lateral, na parte baixa, ou no painel inferior, a escolher.

Cortes

No ***Kit Curso Arduino Automation*** temos apenas uma sugestão de corte para facilitar a retirada do painel do fundo.

Ocorre que quando as tomadas estão instaladas, o painel do fundo não pode ser retirado. Para tal, é necessário desmontar uma das tomadas, afim de extrair o painel. Como isso é extremamente chato de fazer toda hora que precisamos remover o painel, então a sugestão é efetuar um corte, do tamanho de um gomo (o painel é separado por gomos), para que o painel possa ser removido sem a necessidade de desmontar as tomadas.

Vejamos no diagrama abaixo essa sugestão de corte chamada de Área de Corte “AC1”:



Vejamos abaixo a tabela de códigos dos cortes:

Código do Furo	Descrição	Área	Posição
AC1	Corte no painel do fundo para passar pela Tomada T2.	1 gomo do painel	Superior esquerda.

Acessórios

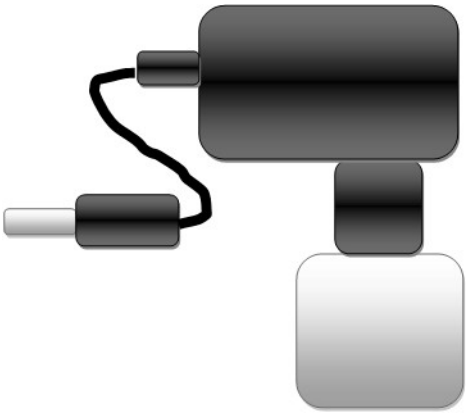
O ***Kit Curso Arduino Automation*** é fornecido com alguns acessórios, dependendo da versão. Vejamos alguns dos principais.

Fonte de Alimentação 12V / 1A

Acessório essencial para o funcionamento da Central de Controle Automation. Duas (2) fontes são incluídas no ***Kit Curso Arduino Automation*** e possuem as seguintes funcionalidades:

- *Fonte 1* – Alimentação geral do circuito; e
- *Fonte 2* – Acionamento da bobina da fechadura eletrônica.

Veja a representação da ***Fonte de Alimentação 12V / 1A*** no diagrama abaixo:



As fontes de alimentação e as tomadas fornecidas no ***Kit Curso Arduino Automation*** são compatíveis com os modelos exigidos pelas Normas brasileiras.

Duas fontes acompanham o ***Kit Curso Arduino Automation***, mas elas também podem ser adquiridas separadamente, no seguinte link:

http://seriallink.com.br/loja/product_info.php?products_id=84

Expansores de Portas Automation

Os expansores de portas são simples adaptadores que permitem a ligação de mais de um fio no mesmo *Borne Parafusável*. As vezes precisamos ligar vários fios no mesmo borne, dificultando as conexões. Com estes adaptadores, teremos apenas um fio por conector Sindal e todos os fios juntos soldados são conectados no borne, que não fica sobrecarregado!

Vejamos a representação desses adaptadores na tabela abaixo:

Expansor de Portas 2 Vias	**Expansor de Portas 3 Vias**

As pontas dos fios são estanhadas juntas, garantindo uma melhor conexão.

No ***Kit Curso Arduino Automation***, são fornecidos os dois modelos citados acima, sendo 1 de cada. Eles podem ser comprados também pelo site:

http://seriallink.com.br/loja/product_info.php?cPath=61&products_id=350

Montagem & Manutenção Geral

Nesta etapa do manual, veremos como montar o ***Kit Curso Arduino Automation*** e também aprenderemos algumas tarefas básicas de manutenção. Maiores detalhes sobre manutenção, no *Manual do Instalador*.

Para começar, vamos ver a lista de ferramentas e equipamentos necessários, depois vamos montar todas as placas, chicotes e rabichos; e só então vamos instalar os programas e carregar nossos códigos nos dois (02) Arduinos Supervisórios! Isso mesmo, temos que carregar dois códigos distintos, um em cada Arduino Supervisório. Relembrando, temos um código para apresentar o *Painel de Controle Web* e outro para controlar os *Módulos Dimmer AC Bivolt*!

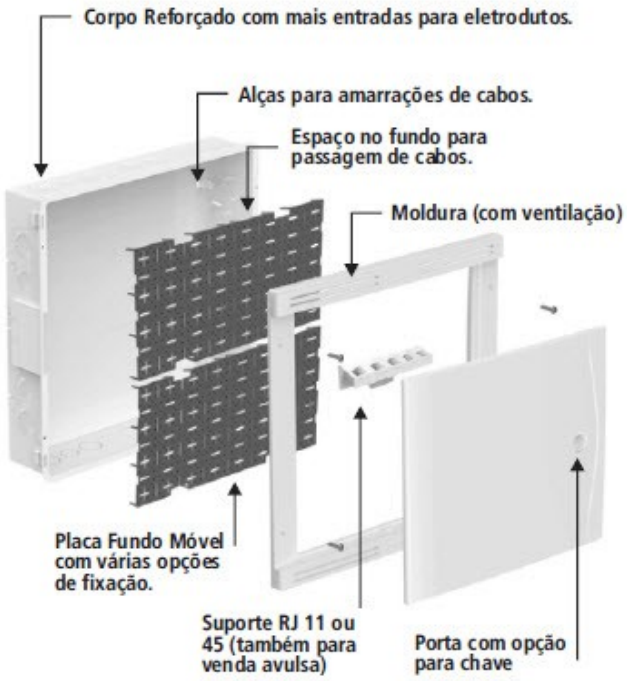
Material Necessário

Para a correta montagem do ***Kit Curso Arduino Automation***, vamos precisar de alguns equipamentos e ferramentas adicionais, que não são fornecidas.

- Chave de Fenda (pequena para os bornes)
- Chave Philips (ou estrela, como também é conhecida – tamanho médio)
- Alicate de Bico
- Alicate de Corte (utilize para cortar as pontas estanhadas muito grandes)
- Furadeira
- Brocas Para Plástico/Madeira
  - 2mm
  - 25mm (broca plana)
  - 12mm (broca plana)
- Brocas de Vídea (para cimento/concreto)
  - 8mm (fixação das buchas na parede)
- Tesoura (para cortar as cintas plásticas)
- Nível de Bolha (para não ficar tudo torto!)
- Punção (para marcar a furação da parede)

Instalação da Caixa VDI

A *Caixa VDI* da *Tigre* é um produto nacional, de qualidade, muito bem resolvido, que atende todas nossas necessidades! Vejamos as características da caixa, pela visão explodida.


Corpo Reforçado com mais entradas para eletrodutos.
Alças para amarrações de cabos.
Espaço no fundo para passagem de cabos.
Moldura (com ventilação)
Placa Fundo Móvel com várias opções de fixação.
Suporte RJ 11 ou 45 (também para venda avulsa)
Porta com opção para chave

Podemos verificar pela imagem, que os itens da caixa são separáveis por:

- Corpo
- Placa Fundo Móvel (Eu chamo de Painel)
- Moldura
- Suporte de Cabos
- Porta

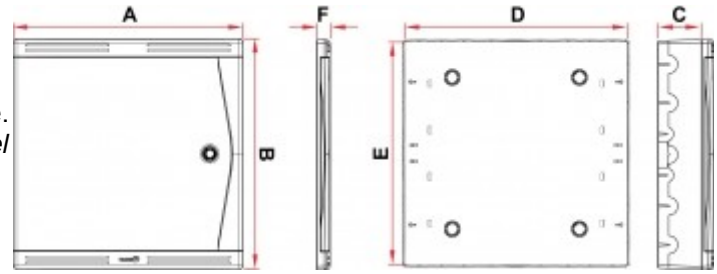
Para nossa montagem, as partes importantes são: *Placa Fundo Móvel*, que chamaremos nesse manual de apenas *Painel*; e o *Corpo*, que utilizaremos para fixar na parede.

Atenção! A *Caixa VDI* fornecida no ***Kit Curso Arduino Automation*** é de *sobrepor na parede*. Caso deseje usar outro tipo ou tamanho, indique isso na sua compra.

Vídeo Institucional da *Tigre*:
http://youtu.be/Lq4lvyfyx7k

Instalando a Caixa VDI na Parede

Bom, primeiro vamos precisar do *Nível de Bolha*, da *Furadeira* e dos parafusos e buchas que acompanham o ***Kit Curso Arduino Automation***. Separe o saquinho que tem 4 buchas de 8 mm e 4 parafusos compatíveis.



Repare no diagrama que o Corpo tem em seu fundo quatro furos para fixação. Vamos utilizar essa furação para fazer as marcações na parede. Utilize a punção para isso. Utilize também o *Nível de Bolha* para alinhar corretamente a *Caixa VDI* na parede.

Atenção! Esta furação do fundo do corpo não possui código, pois é original da *Tigre*, *não* é da Serial Link! Lembre-se, as furações deste manual que possuem código, são *sugestões* feitas pela *Serial Link*. Os furos do fundo da caixa são originais, feitos na fabricação, pela *Tigre*.

Marque, fure e fixe a caixa na parede. Pode fazer isso agora, pois precisaremos da caixa no lugar para que possamos planejar a fiação do cômodo, e deixar ela pronta.

Não se preocupe com a montagem das placas, pois o Painel do Fundo é removível e trabalharemos fora da caixa. Precisaremos do *Painel* no lugar para medir a localização das Tomadas T1 e T2, do *Chicote "A"*.

Furação e Corte da Caixa VDI

Conforme vimos acima, neste manual, precisamos efetuar alguns furos e cortes adicionais.

Furação

Para seguir as sugestões de furos deste manual, siga os passos:

- Com a Caixa VDI corretamente instalada na parede, deixe o Painel de Fundo no lugar e com o Chicote “A” nas mãos, vamos medir a furação das Tomadas T1 e T2.
- A Tomada T1 vai localizada na borda inferior da caixa, no canto direito, conforme já vimos nos diagramas de Furação; e correspondem aos furos FA1 e FA2
- A Tomada T2 vai localizada na borda superior da caixa, no canto esquerdo, conforme já vimos nos diagramas de Furação; e correspondem aos furos FB1 e FB2

Atenção! Faça a marcação dos furos, utilizando as tomadas como referência, mas não fure diâmetros maiores que 3mm senão os parafusos vão ficar frouxos e não vão fixar a tomada corretamente. É importante que o furo tenha por volta de 2mm para que o próprio parafuso termine de abrir o furo, conforme for sendo parafusado.

O correto posicionamento das Tomadas T1 e T2 só é possível com o Painel de Fundo no lugar, pois utilizamos ele para limitar a profundidade das tomadas em relação a borda.

Se fixarmos as tomadas sem o Painel de Fundo, depois verificaremos que não é possível encaixar de volta o Painel. Por isso fazemos dessa forma.

Aproveite para efetuar os furos FC1, FD1 e FE1. Utilize o Furo “FE1” para passar o rabicho do Chicote “A1” que liga no Chicote “A2”.

Os Furos “FC1” e “FD1” serão utilizados mais adiante, em outra etapa da montagem.

Atenção! Cuidado ao furar a Caixa VDI da Tigre! O material não é elástico, pelo contrário, costuma trincar e rachar quando furado. Muito cuidado principalmente no furo de 25mm para não quebrar a caixa!

Cortes

O corte sugerido neste manual é apenas um e garante a remoção do Painel de Fundo, após as Tomadas T1 e T2 terem sido instaladas.

O instalador nesse momento pode optar por não fazer esse corte, mas tenha em mente que será quase impossível retirar o Painel de Fundo, sem desmontar pelo menos uma das tomadas.

Verifique o *Diagrama de Cortes* para saber onde cortar o *Painel de Fundo* para que ele possa ser removido facilmente.

Montagem das Placas no Painel

Neste momento da montagem, você pode optar por trabalhar com o *Painel de Fundo* montado na *Caixa VDI* ou você pode retirá-lo, trabalhar na mesa (ou bancada), e depois recolocar o *Painel de Fundo* já com as placas montadas.

Utilize as cintas plásticas fornecidas juntamente com o ***Kit Curso Arduino Automation*** para prender as placas e módulos no *Painel de Fundo*.

Atenção! Nos DVD's que você recebeu ao adquirir o ***Kit Curso Arduino Automation*** também possuem muitas dicas e exemplos de utilização das cintas. Na verdade todo o processo de montagem está gravado em vídeo, caso o instalador queira acompanhar a instalação assistindo as aulas.

Neste documento apresentamos a mesma organização de placas, sobre o *Painel de Fundo*, que foram feitas no curso. Vejamos abaixo o diagrama de sugestão:

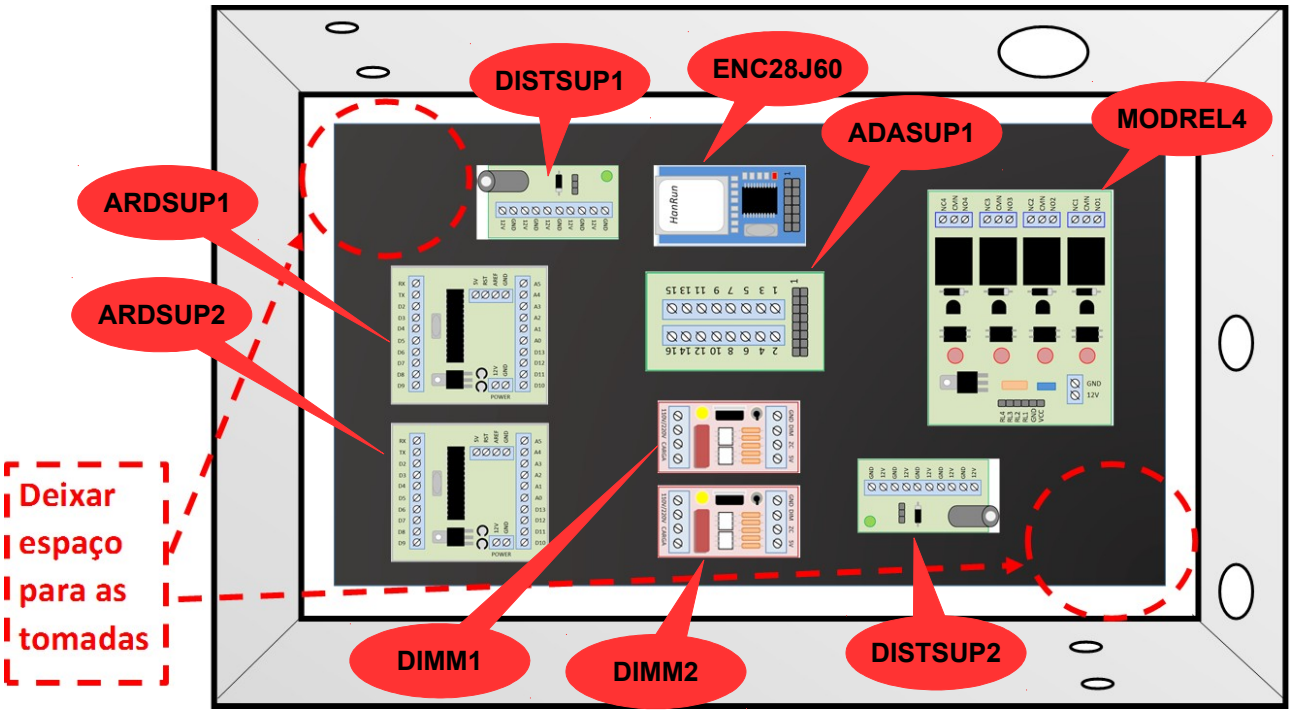


Tabela de identificação das placas do diagrama acima:

Código	Descrição	Orientação
DISTSUP1	Distribuidor de Alimentação Supervisório da Fechadura	Conector J4 voltado para esquerda.
ENC28J60	Módulo Ethernet ENC28J60	Deixar espaço para o Cabo de Rede!
ADASUP1	Adaptador Pinos/Bornes Supervisório	Indiferente
MODREL4	Módulo de Relê 4 Canais	Deixe sempre os relês voltados para cima, alinhados com o Furo “FC1”
DISTSUP2	Distribuidor de Alimentação Supervisório Geral	Conector J4 voltado para direita.

DIMM2	Módulo Dimmer AC Bivolt 2	110/220V/Carga voltado para direita.
DIMM1	Módulo Dimmer AC Bivolt 1	110/220V/Carga voltado para direita.
ARDSUP2	Arduino Supervisório 2	Conector de Bornes chamado Power, voltado para baixo.
ARDSUP1	Arduino Supervisório 1	Conector de Bornes chamado Power, voltado para baixo.

Não esqueça de deixar espaço para as tomadas. Calcule o espaço necessário, utilizando a *Caixa VDI* e o *Chicote "A"* que você já montou na etapa anterior. Não é fácil acertar de primeira, faça umas experiências para melhor acomodação das placas no *Painel de Fundo*.

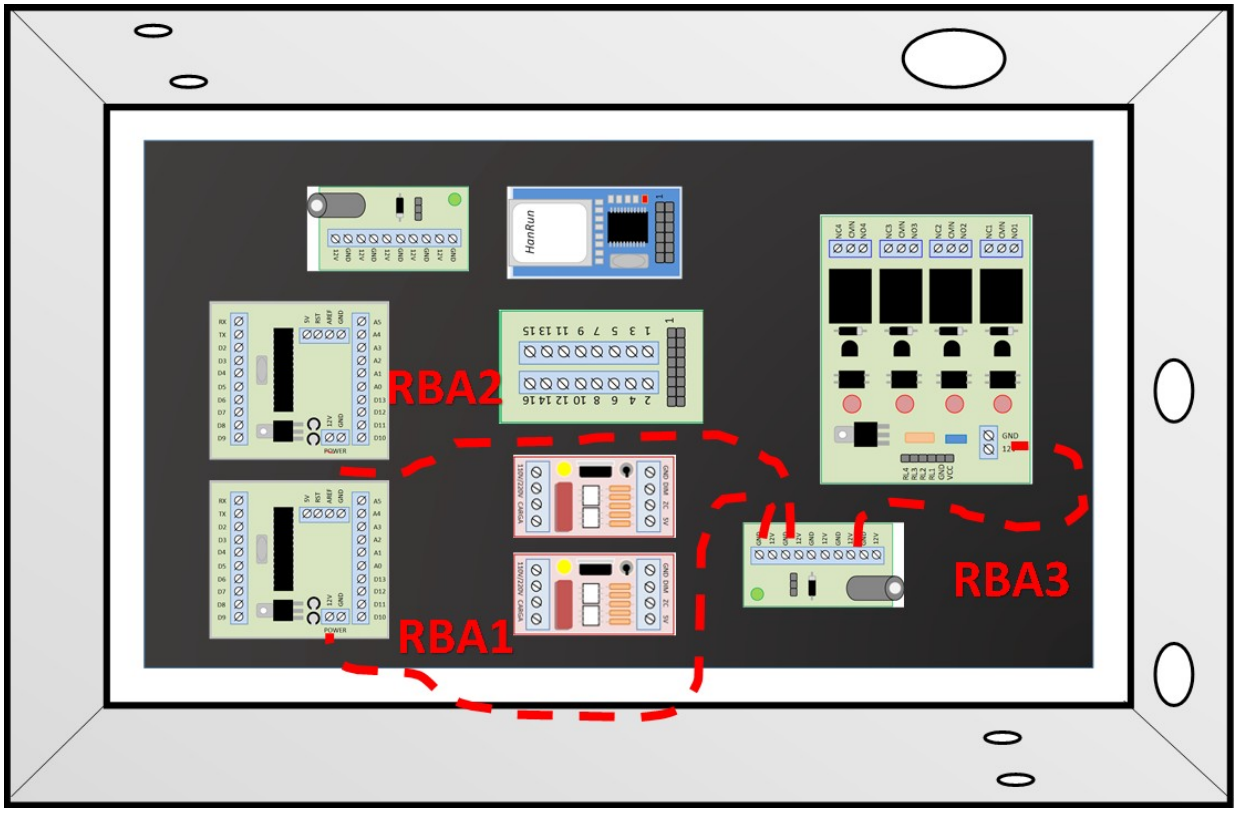
Diagramas de Conexões dos Rabichos

Vamos então começar, pra valer, a efetuar as conexões do ***Kit Curso Arduino Automation***! Lembrando que temos 4 tipos de *Rabichos*:

- Alimentação;
- Comunicação;
- Conexão (Macho e Fêmea); e
- Especial

Rabichos de Alimentação

Verifique no esquema abaixo a localização dos três (03) *Rabichos de Alimentação*. Conforme já vimos, estes rabichos são do mesmo tamanho (30cm) e possuem apenas dois (02) fios, um vermelho e um preto.



Vejamos as tabelas de conexão para cada Rabicho de Alimentação!

Rabicho “RBA1”

Terminal Origem	Placa Origem	Cor do Fio
12V	Distribuidor de Alimentação Supervisório 1	Vermelho
GND	Distribuidor de Alimentação Supervisório 1	Preto

Terminal Destino	Placa Destino	Cor do Fio
12V	Arduino Supervisório 2	Vermelho
GND	Arduino Supervisório 2	Preto

Rabicho “RBA2”

Terminal Origem	Placa Origem	Cor do Fio
12V	Distribuidor de Alimentação Supervisório 1	Vermelho
GND	Distribuidor de Alimentação Supervisório 1	Preto

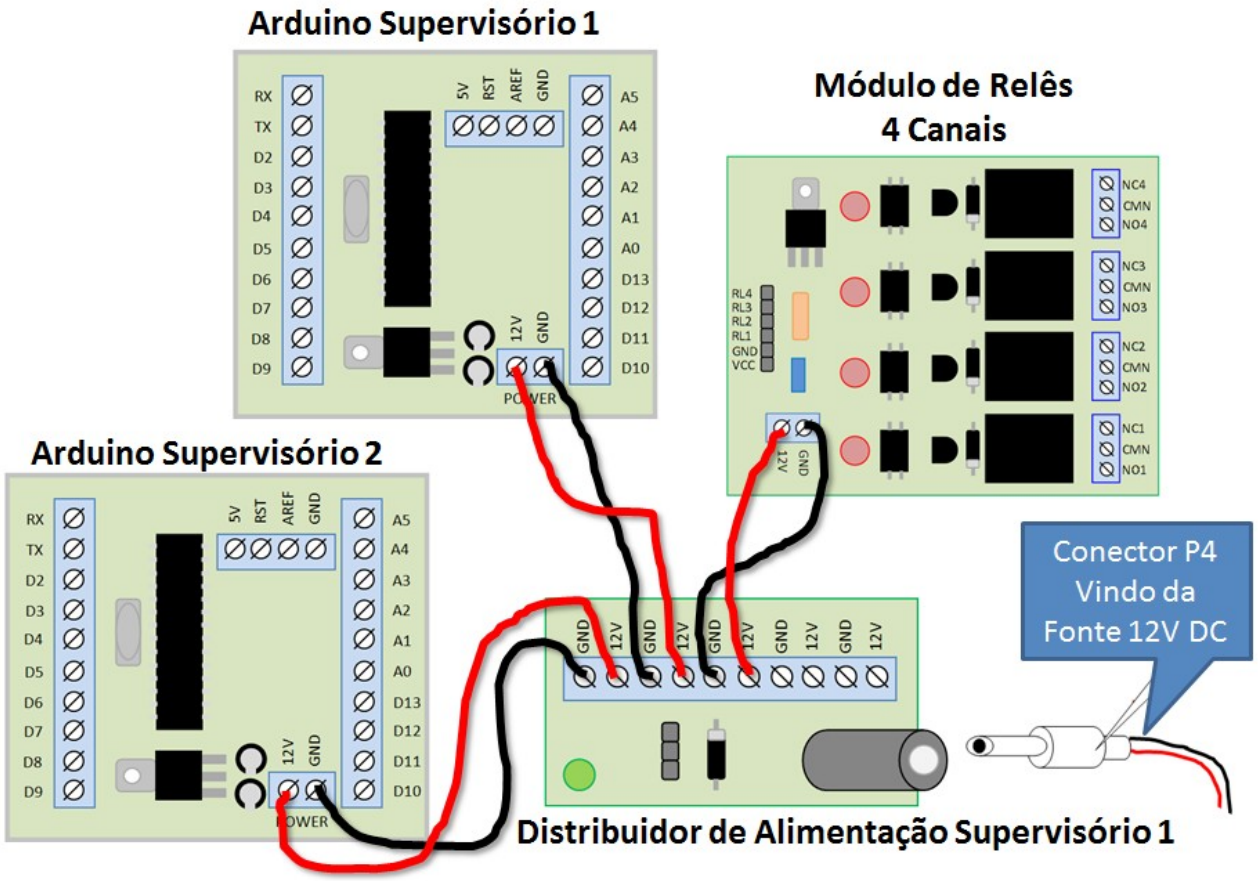
Terminal Destino	Placa Destino	Cor do Fio
12V	Arduino Supervisório 1	Vermelho
GND	Arduino Supervisório 1	Preto

Rabicho “RBA3”

Terminal Origem	Placa Origem	Cor do Fio
12V	Distribuidor de Alimentação Supervisório 1	Vermelho
GND	Distribuidor de Alimentação Supervisório 1	Preto

Terminal Destino	Placa Destino	Cor do Fio
12V	Módulo de Relê 4 Canais	Vermelho
GND	Módulo de Relê 4 Canais	Preto

Confira agora o esquema de ligação descrito nas tabelas acima:

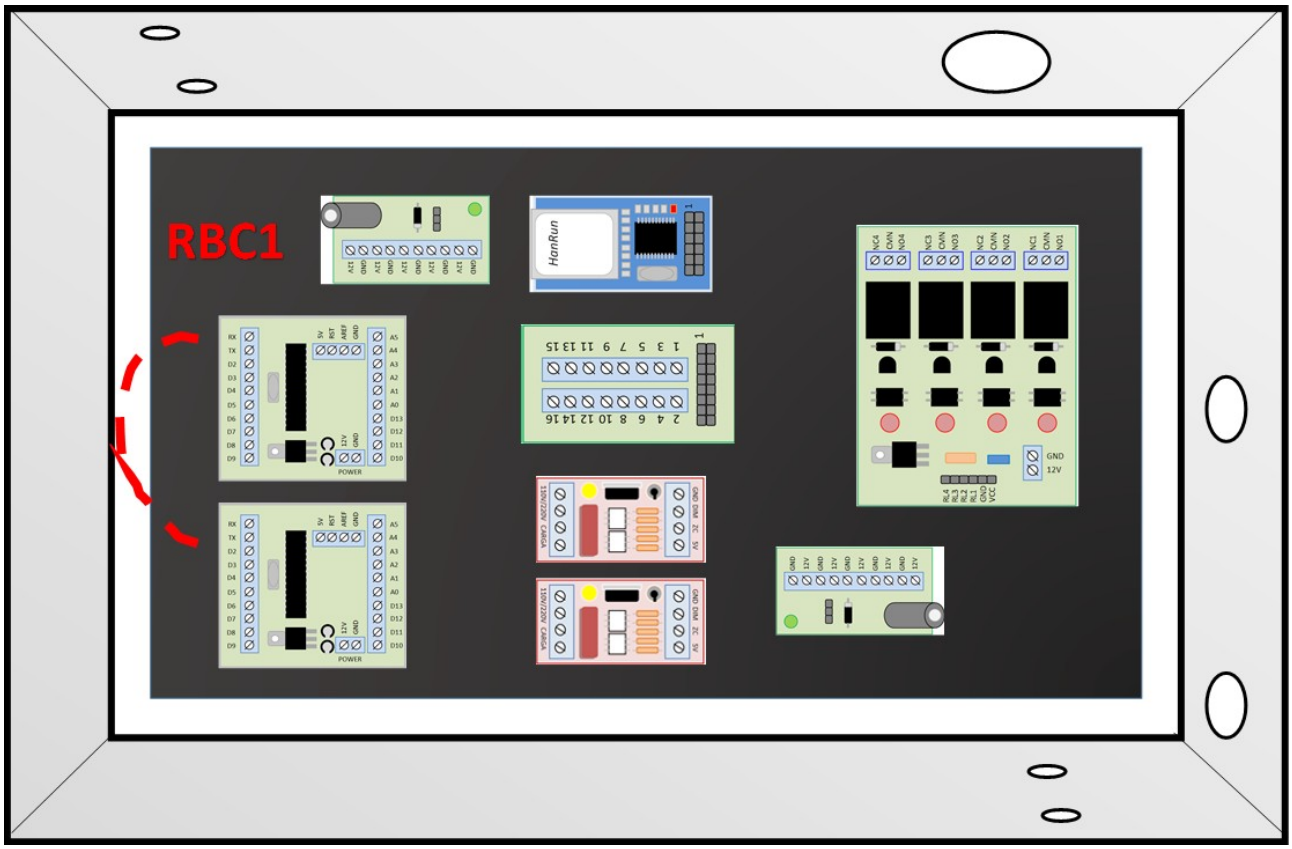


Rabichos de Comunicação

Temos apenas um *Rabicho de Comunicação Serial* no ***Kit Curso Arduino Automation***, que interliga o *Arduino Supervisório 1* com o *Arduino Supervisório 2*.

Este *Rabicho "RBC1"* é responsável por transmitir os comandos pela *Comunicação Serial* que controlam os *Módulos Dimmer AC Bivolt*.

Vejamos a localização desse rabicho em relação a caixa, no diagrama abaixo:



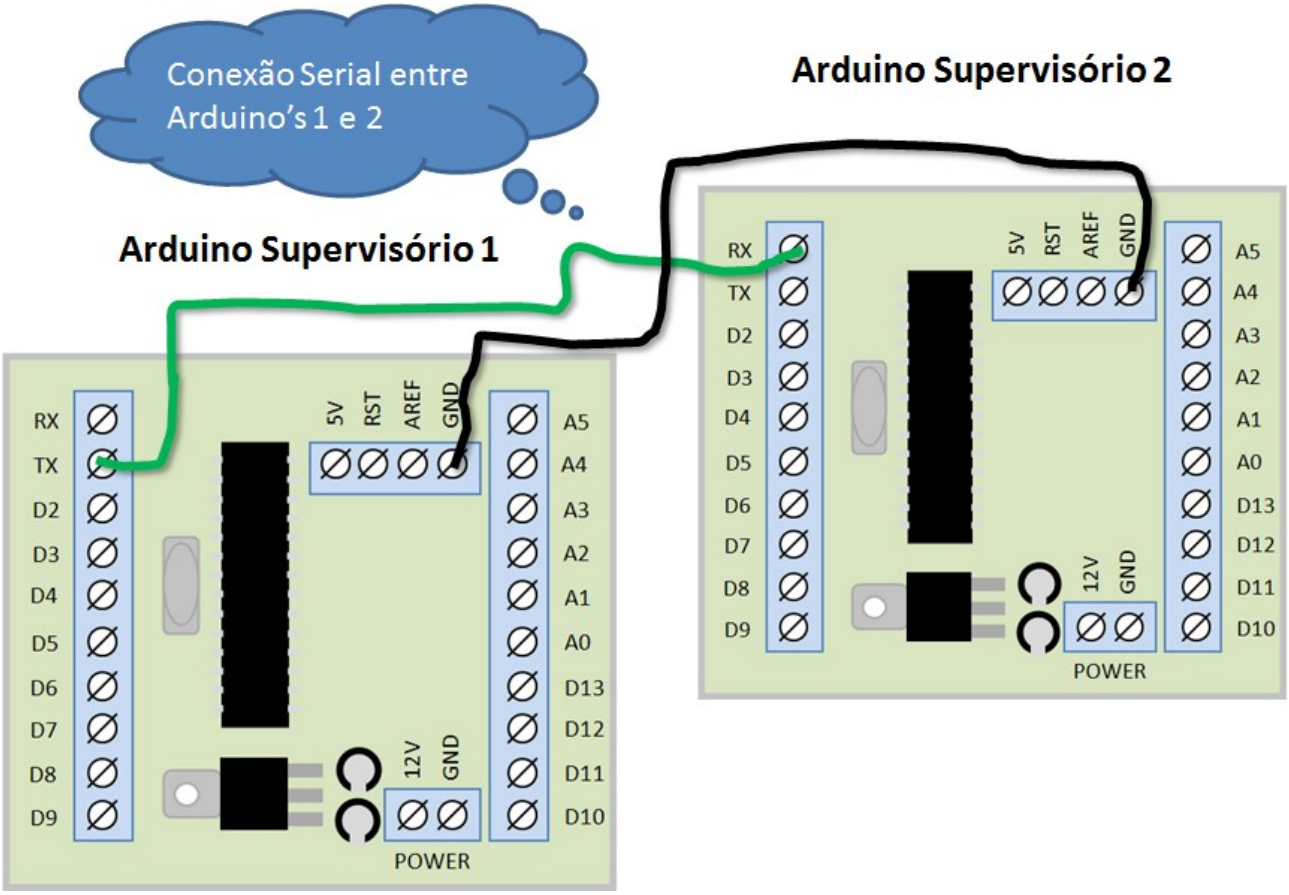
Vejamos as tabelas de conexão para os Rabichos de Comunicação!

Rabicho "RBC1"

Terminal Origem	Placa Origem	Cor do Fio
TX	Arduino Supervisório 1	Verde
GND	Arduino Supervisório 1	Preto

Terminal Destino	Placa Destino	Cor do Fio
RX	Arduino Supervisório 2	Verde
GND	Arduino Supervisório 2	Preto

Confira agora o esquema de ligação descrito nas tabelas acima:


Conexão Serial entre Arduino's 1 e 2
Arduino Supervisório 1
Arduino Supervisório 2
RX
TX
D2
D3
D4
D5
D6
D7
D8
D9
5V
RST
AREF
GND
12V
GND
POWER
A5
A4
A3
A2
A1
A0
D13
D12
D11
D10

Rabichos de Conexão

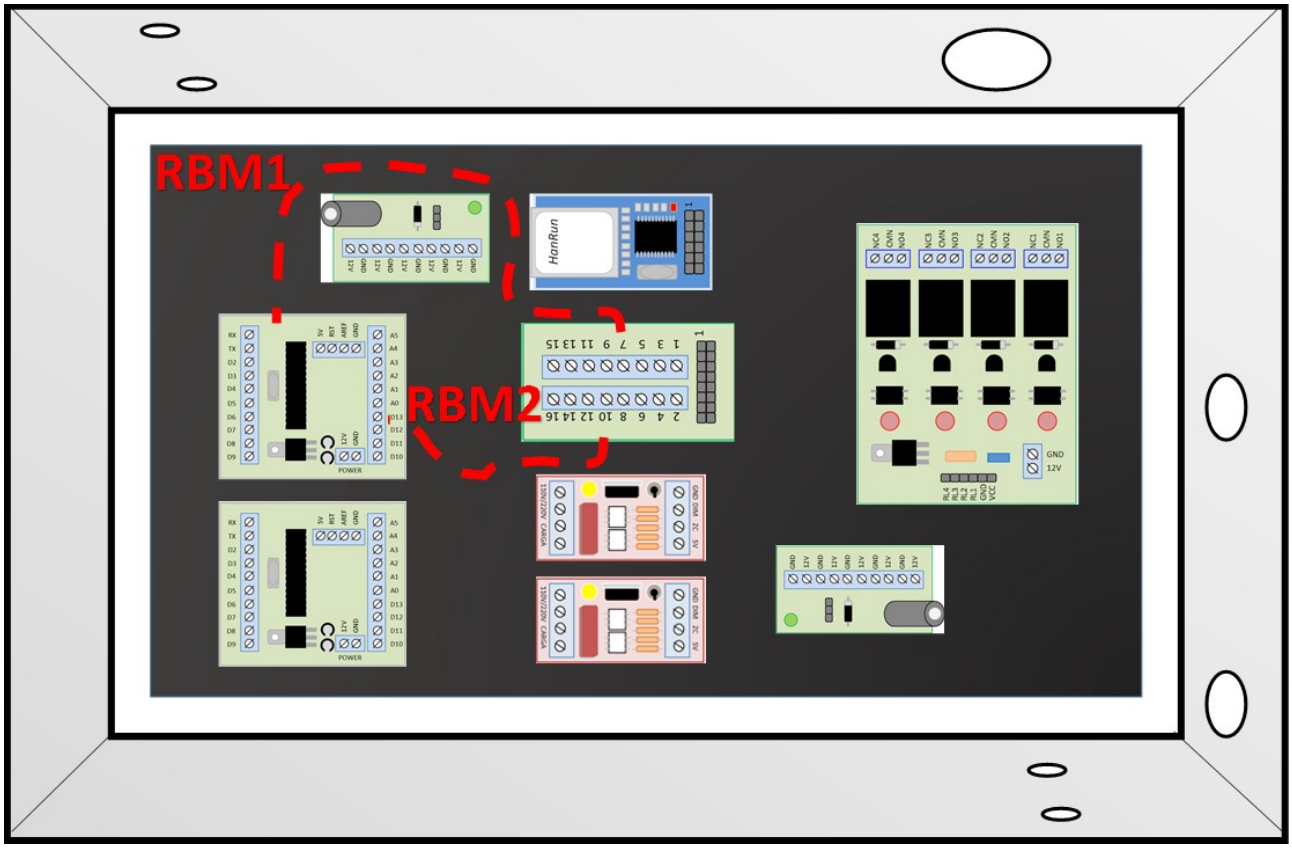
Já vimos anteriormente neste manual que os Rabichos de Conexão são separados em dois sub-grupos. São os:

- Machos: fios sem terminais, apenas com as pontas estanhadas; e
- Fêmeas: fios com terminais MODU fêmea em ambas as pontas.

Rabichos de Conexão Machos

Este tipo de Rabicho é utilizado para outras placas com o ***Adaptador de Pinos/Bornes Supervisório***, na sua parte de *bornes*.

Vejamos a localização desses rabichos em relação a caixa, no diagrama abaixo:



Vejamos as tabelas de conexão para os Rabichos de Conexão Machos!

Rabicho “RBM1”

Terminal Origem	Placa Origem	Cor do Fio
GND	Arduino Supervisório 1	Preto
D5	Porta Digital 5 - Arduino Supervisório 1	Azul
D6	Porta Digital 6 - Arduino Supervisório 1	Amarelo
D7	Porta Digital 7 - Arduino Supervisório 1	Verde
D8	Porta Digital 8 - Arduino Supervisório 1	Branco

Terminal Destino	Placa Destino	Cor do Fio
16	Adaptador de Pinos/Bornes Supervisório	Preto
15	Adaptador de Pinos/Bornes Supervisório	Azul
13	Adaptador de Pinos/Bornes Supervisório	Amarelo
11	Adaptador de Pinos/Bornes Supervisório	Verde
9	Adaptador de Pinos/Bornes Supervisório	Branco

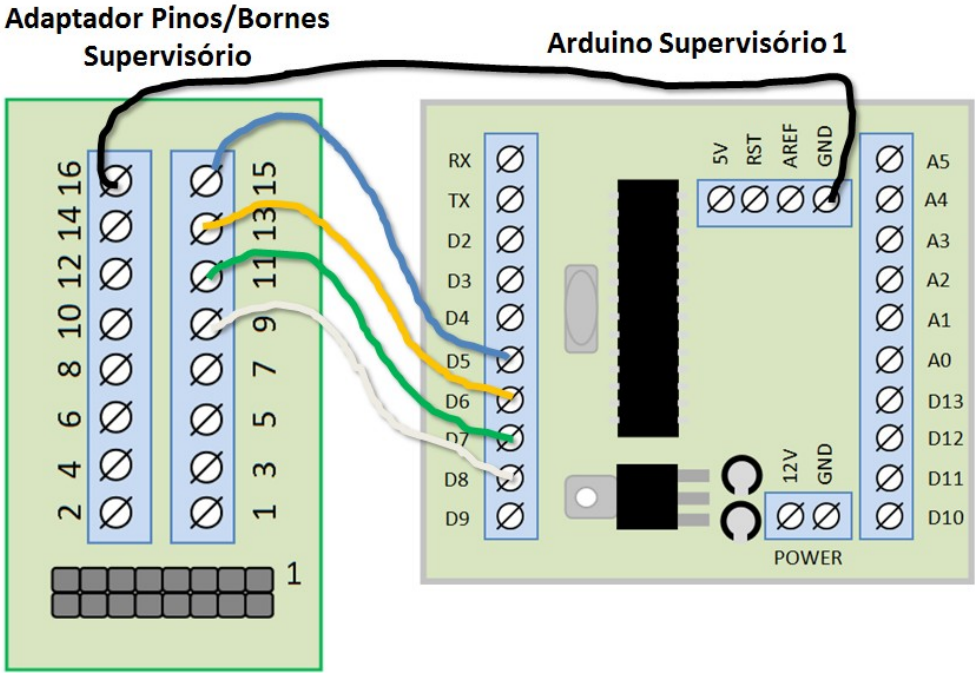
Rabicho “RBM2”

Terminal Origem	Placa Origem	Cor do Fio
GND	Arduino Supervisório 1	Azul
D10	Porta Digital 10 - Arduino Supervisório 1	Branco
D11	Porta Digital 11 - Arduino Supervisório 1	Verde
D12	Porta Digital 12 - Arduino Supervisório 1	Amarelo
D13	Porta Digital 13 - Arduino Supervisório 1	Preto
5V	Arduino Supervisório 1	Vermelho

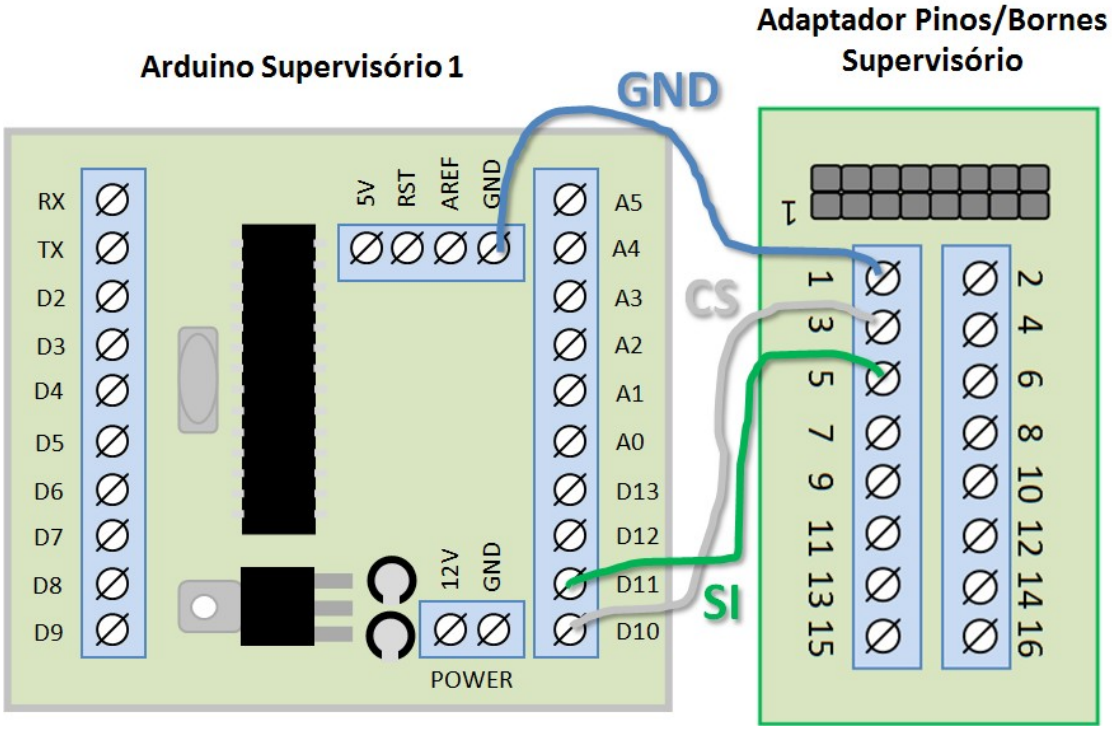
Terminal Destino	Placa Destino	Cor do Fio
1	Adaptador de Pinos/Bornes Supervisório	Azul
3	Adaptador de Pinos/Bornes Supervisório	Branco
5	Adaptador de Pinos/Bornes Supervisório	Verde
10	Adaptador de Pinos/Bornes Supervisório	Amarelo
8	Adaptador de Pinos/Bornes Supervisório	Preto
12	Adaptador de Pinos/Bornes Supervisório	Vermelho

Confira agora o esquema de ligação descrito nas tabelas acima:

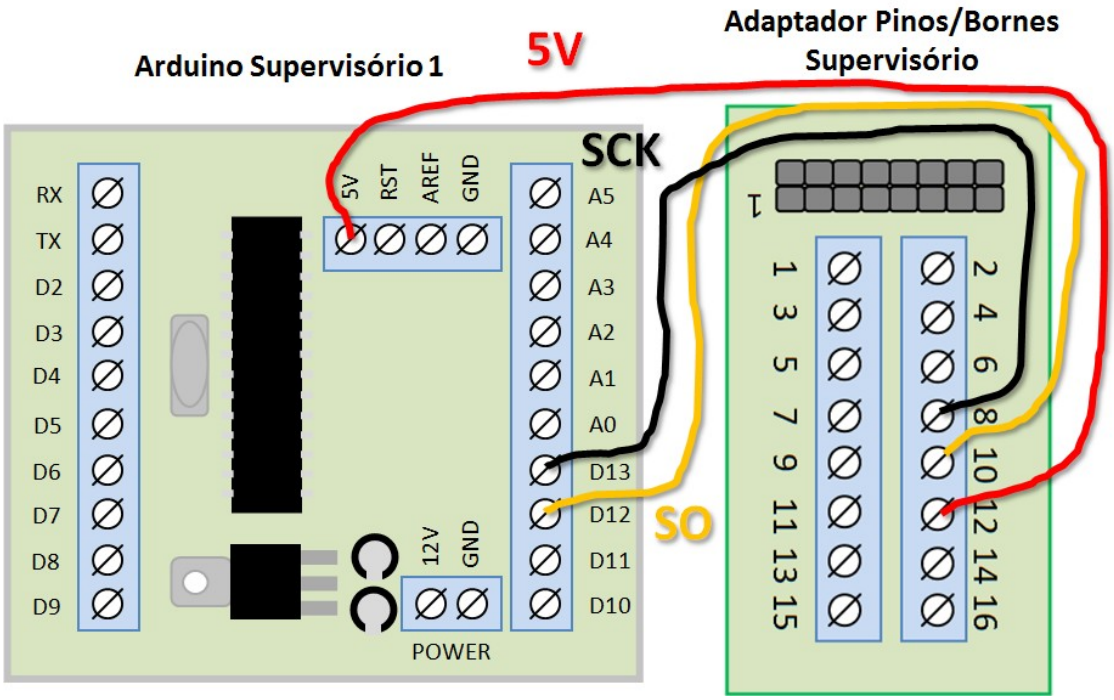
RBM1 – Parte 1/1



RBM2 – Parte 1/2



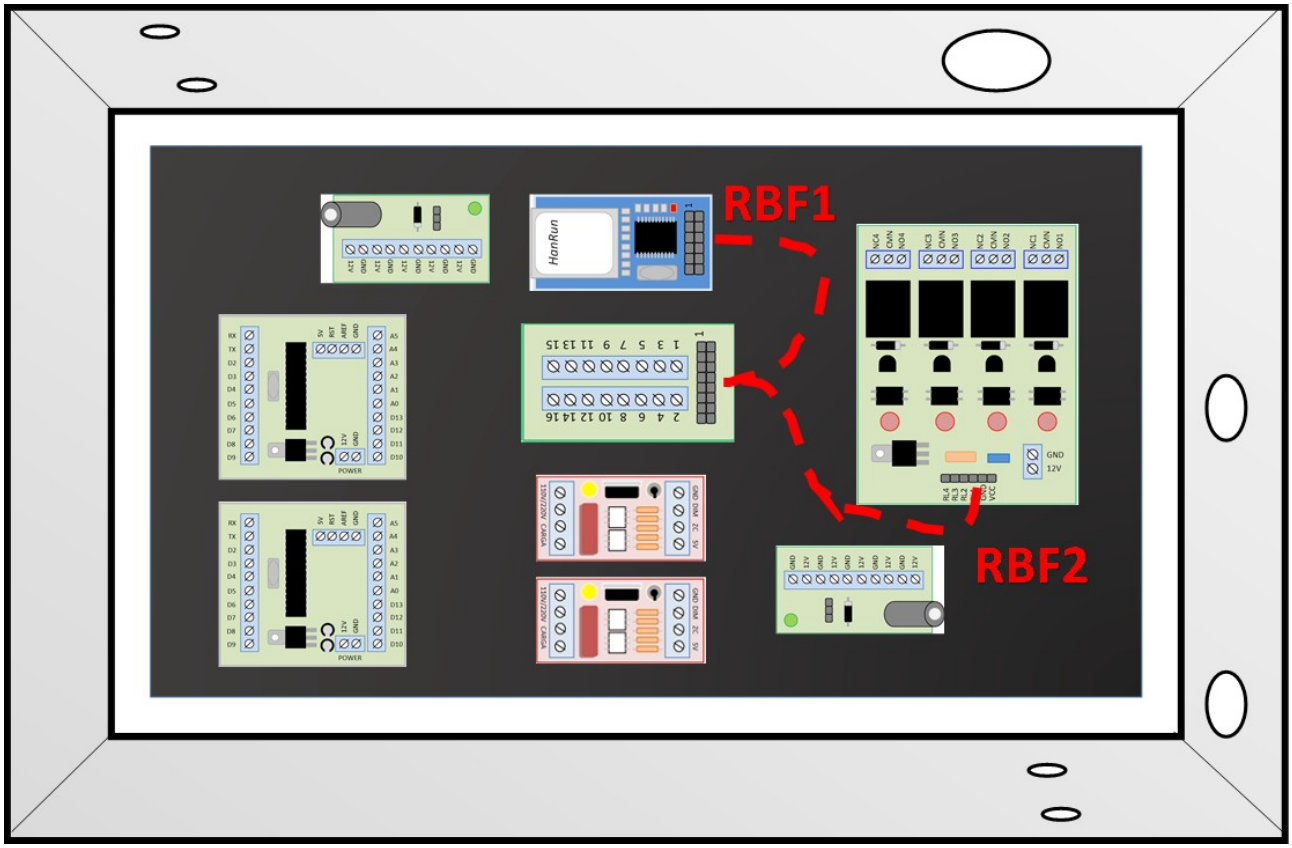
RBM2 – Parte 2/2



Rabichos de Conexão Fêmeas

Este tipo de *Rabicho* é utilizado para outras placas com o ***Adaptador de Pinos/Bornes Supervisório***, na sua parte de *pinos*.

Vejamos a localização desses rabichos em relação a caixa, no diagrama abaixo:



Vejamos as tabelas de conexão para os Rabichos de Conexão Fêmeas!

Rabicho “RBF1”

Terminal Origem	Placa Origem	Cor do Fio
1	Adaptador de Pinos/Bornes Supervisório	Azul
3	Adaptador de Pinos/Bornes Supervisório	Branco
5	Adaptador de Pinos/Bornes Supervisório	Verde
10	Adaptador de Pinos/Bornes Supervisório	Amarelo
8	Adaptador de Pinos/Bornes Supervisório	Preto
12	Adaptador de Pinos/Bornes Supervisório	Vermelho

Terminal Destino	Placa Destino	Cor do Fio
GND	Módulo Ethernet ENC28J60	Azul
CS	Módulo Ethernet ENC28J60	Branco
SI	Módulo Ethernet ENC28J60	Verde
SO	Módulo Ethernet ENC28J60	Amarelo
SCK	Módulo Ethernet ENC28J60	Preto
5V	Módulo Ethernet ENC28J60	Vermelho

Atenção – O Módulo Ethernet ENC28J60 (na presente revisão deste manual) não é fabricado pela Serial Link, portanto não segue o padrão. Os terminais de destino da tabela acima indicam a etiqueta do pino e não seu número, pois podem existir variações. Procure as ligações pelas etiquetas impressas na placa e não pela numeração.

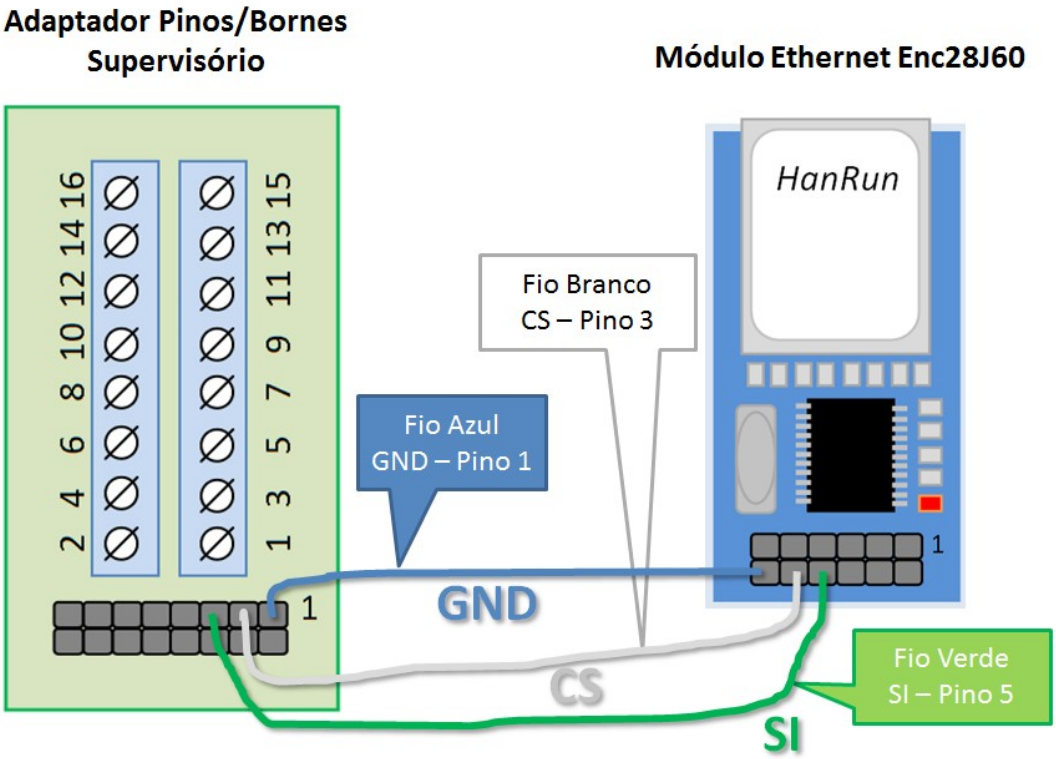
Rabicho “RBF2”

Terminal Origem	Placa Origem	Cor do Fio
GND	Módulo de Relês 4 Canais	Preto
RL1	Módulo de Relês 4 Canais – Canal 1	Azul
RL2	Módulo de Relês 4 Canais – Canal 2	Amarelo
RL3	Módulo de Relês 4 Canais – Canal 3	Verde
RL4	Módulo de Relês 4 Canais – Canal 4	Branco

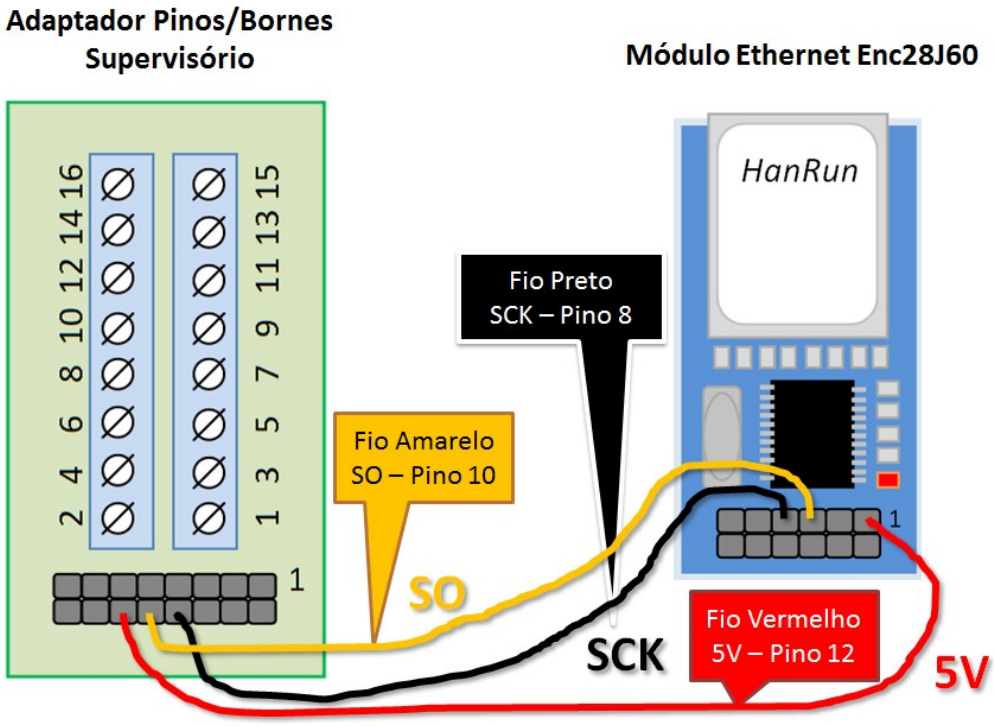
Terminal Destino	Placa Destino	Cor do Fio
16	Adaptador de Pinos/Bornes Supervisório	Preto
15	Adaptador de Pinos/Bornes Supervisório	Azul
13	Adaptador de Pinos/Bornes Supervisório	Amarelo
11	Adaptador de Pinos/Bornes Supervisório	Verde
9	Adaptador de Pinos/Bornes Supervisório	Branco

Confira agora o esquema de ligação descrito nas tabelas acima:

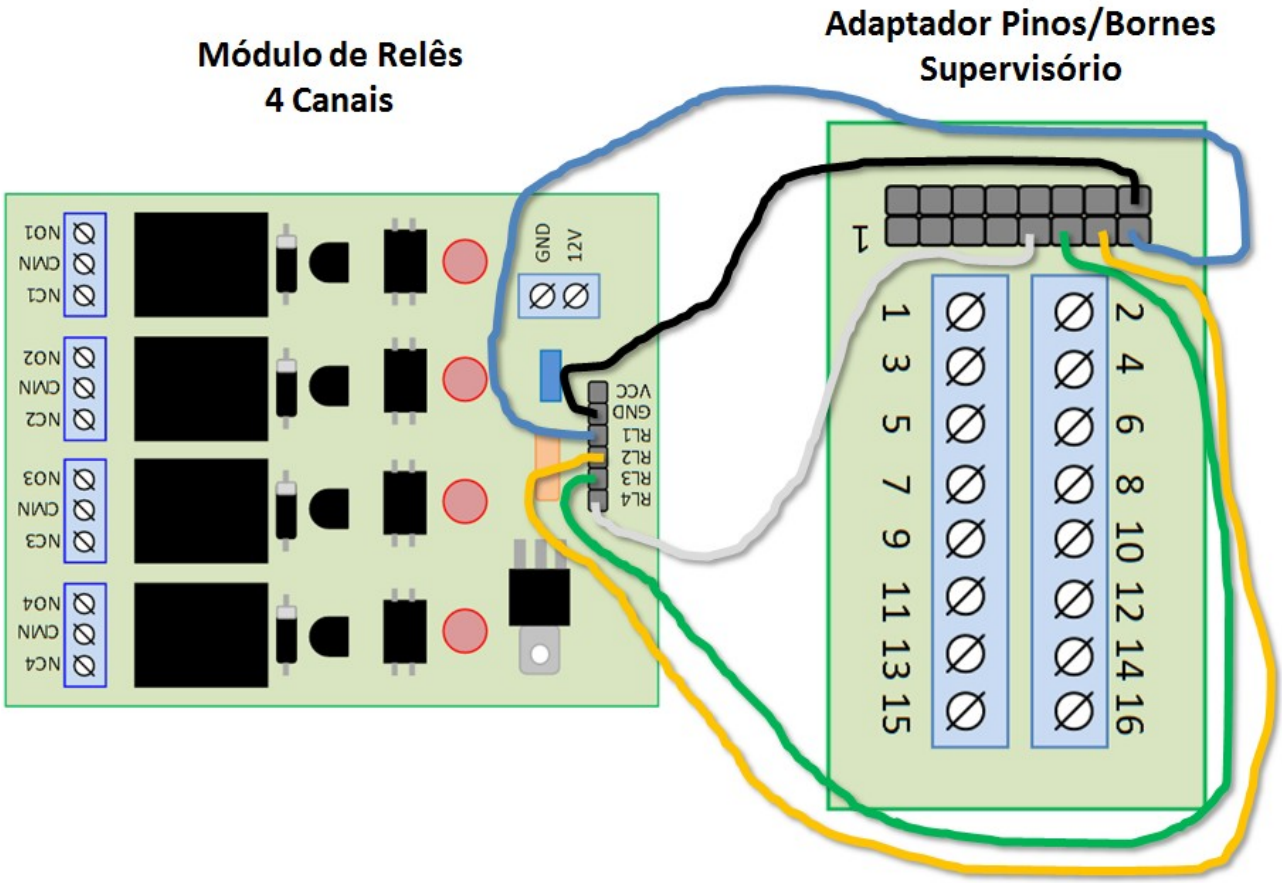
RBF1 – Parte 1/2



RBF1 – Parte 2/2



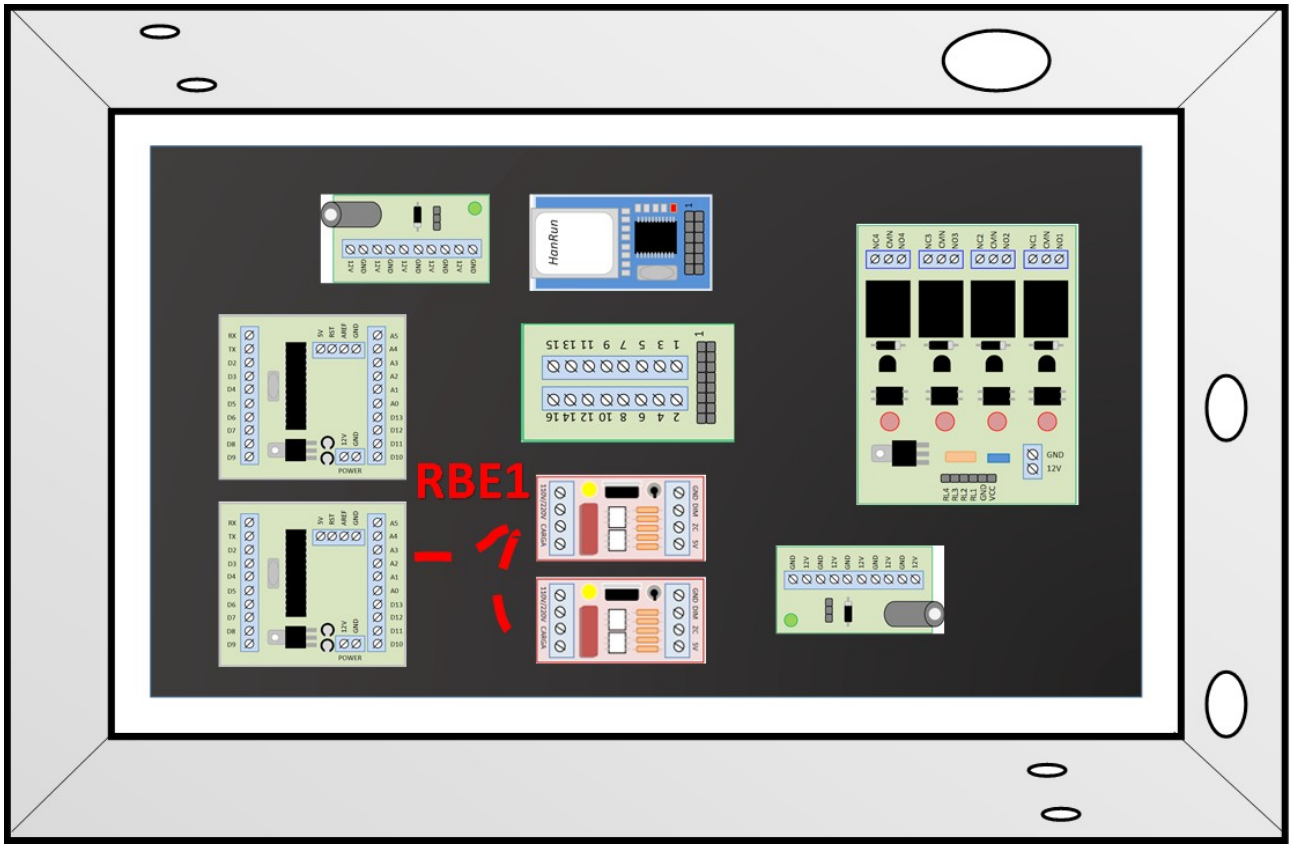
RBF2 – Parte 1/1



Rabichos Especiais

Temos apenas um *Rabicho Especial no Kit*, que efetua a ligação entre o *Arduino Supervisório 2* e os *Módulos Dimmer AC Bivolt*.

Vejamos a localização desse rabicho em relação a caixa, no diagrama abaixo:



Vejamos as tabelas de conexão para o Rabicho Especial dos Dimmers!

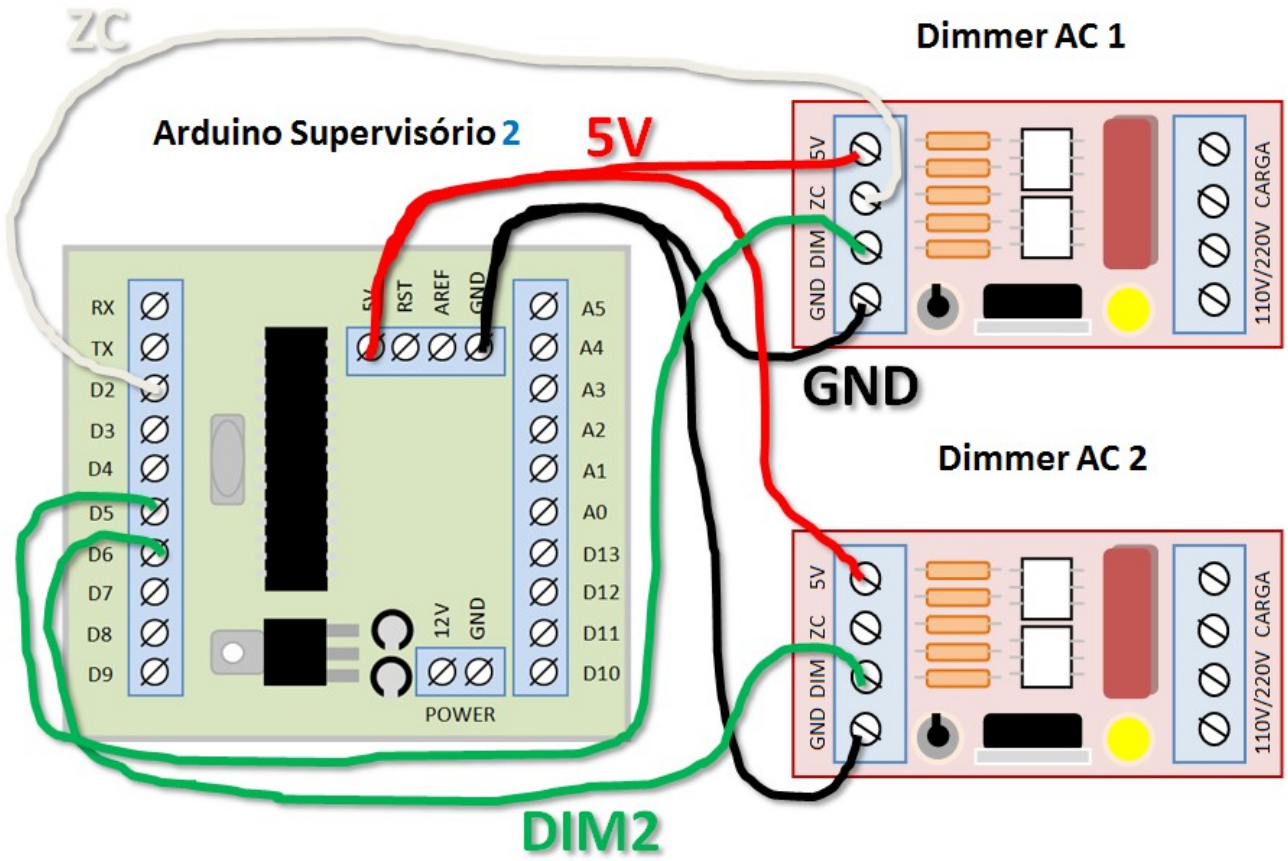
Rabicho “RBE1”

Terminal Origem	Placa Origem	Cor do Fio
GND	Arduino Supervisório 2	Preto
5V	Arduino Supervisório 2	Vermelho
D2	Porta Digital 2 - Arduino Supervisório 2	Branco
D5	Porta Digital 5 - Arduino Supervisório 2	Verde
D6	Porta Digital 6 - Arduino Supervisório 2	Verde

Terminal Destino (Ponta)	Placa Destino	Cor do Fio
GND	Módulo Dimmer AC Bivolt 2	Preto
DIM	Módulo Dimmer AC Bivolt 2	Verde
5V	Módulo Dimmer AC Bivolt 2	Vermelho

Terminal Destino (Emenda)	Placa Destino	Cor do Fio
GND	Módulo Dimmer AC Bivolt 1	Preto
DIM	Módulo Dimmer AC Bivolt 1	Verde
ZC	Módulo Dimmer AC Bivolt 1	Branco
5V	Módulo Dimmer AC Bivolt 1	Vermelho

Confira agora o esquema de ligação descrito nas tabelas acima:



Pronto! Temos todos os Rabichos conectados, podemos partir para a etapa de conexão dos Chicotes!

Diagramas de Conexões dos Chicotes

Como já pudemos verificar anteriormente neste manual, alguns esquemas da documentação antiga traziam códigos diferentes das versões atuais dos *Chicotes*. Vejamos o que mudou para o lançamento do ***Kit Curso Arduino Automation***:

- ~~**Chicote B**~~ – Não será lançado! Na nova versão o *FTDi* conecta diretamente, *sem necessidade de chicote*, no *Arduino Supervisório REV2*
- ~~**Chicote E**~~ – Foi desmembrado! Na nova versão, foi incorporado aos *Chicotes "C"* e *"A"* (a parte de alimentação foi para o *Chicote "A"* e a parte de saída, foi para o *Chicote "C"*)

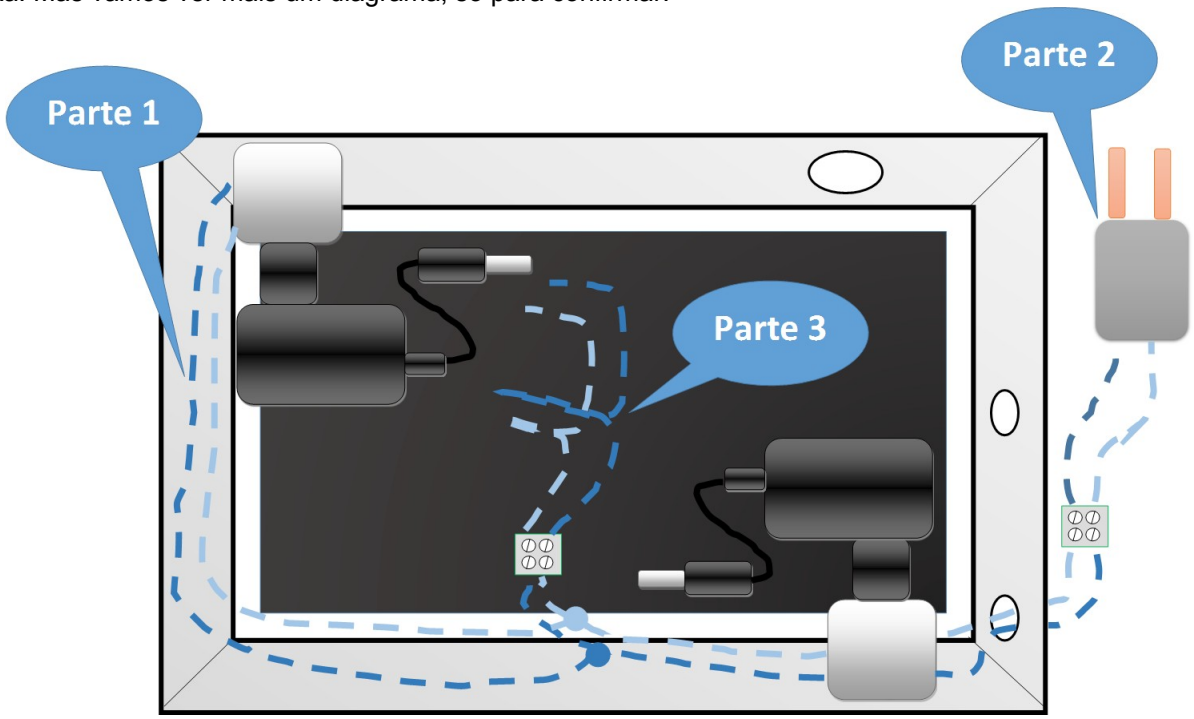
Dessa forma, a configuração da *Nova Versão de Lançamento* do ***Kit Curso Arduino Automation*** , os *Chicotes* ficaram assim:

- **Chicote A** – Toda parte de alimentação, incluindo a parte dos *Módulos Dimmers AC*, conhecido agora por ***Chicote de Alimentação Automation***
- **Chicote C** – Toda parte da saída, conhecido agora por ***Chicote Principal Automation***
- **Chicote D** – Toda parte de acionamento da fechadura, e ficou conhecido por ***Chicote Fechadura 12V Automation***

Vamos então para os diagramas de conexão!

Chicote A – Chicote de Alimentação Automation

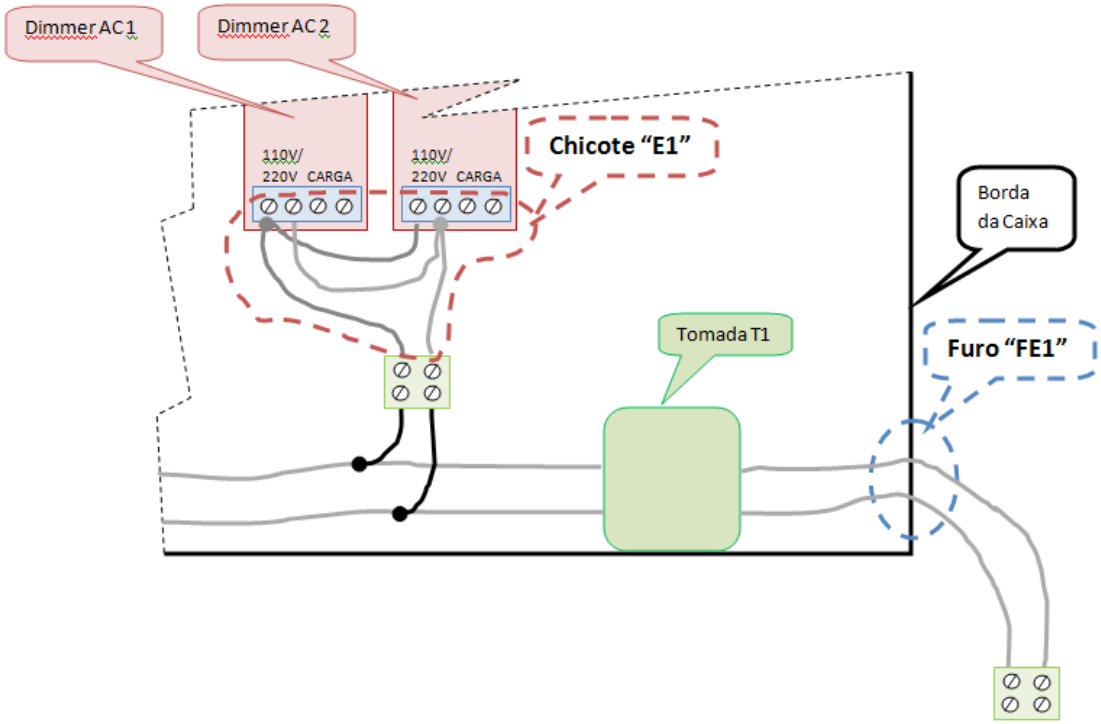
Neste ponto da montagem, o instalador já deve ter furado e fixado esse *Chicote*! Pois já vimos diagramas suficientes para saber que, sem esse chicote montado, não é possível medir as outras peças que vão na caixa. Mas vamos ver mais um diagrama, só para confirmar:



Vejamos então uma lista com as partes desse chicote:

- *Parte 1* – Composta pela fiação principal e duas tomadas
- *Parte 2* – Composta por um rabicho de fio paralelo e um Plugue de Tomada
- *Parte 3* – Composta de um rabicho com emenda para ligar os dois *Módulos Dimmers AC Bivolt*

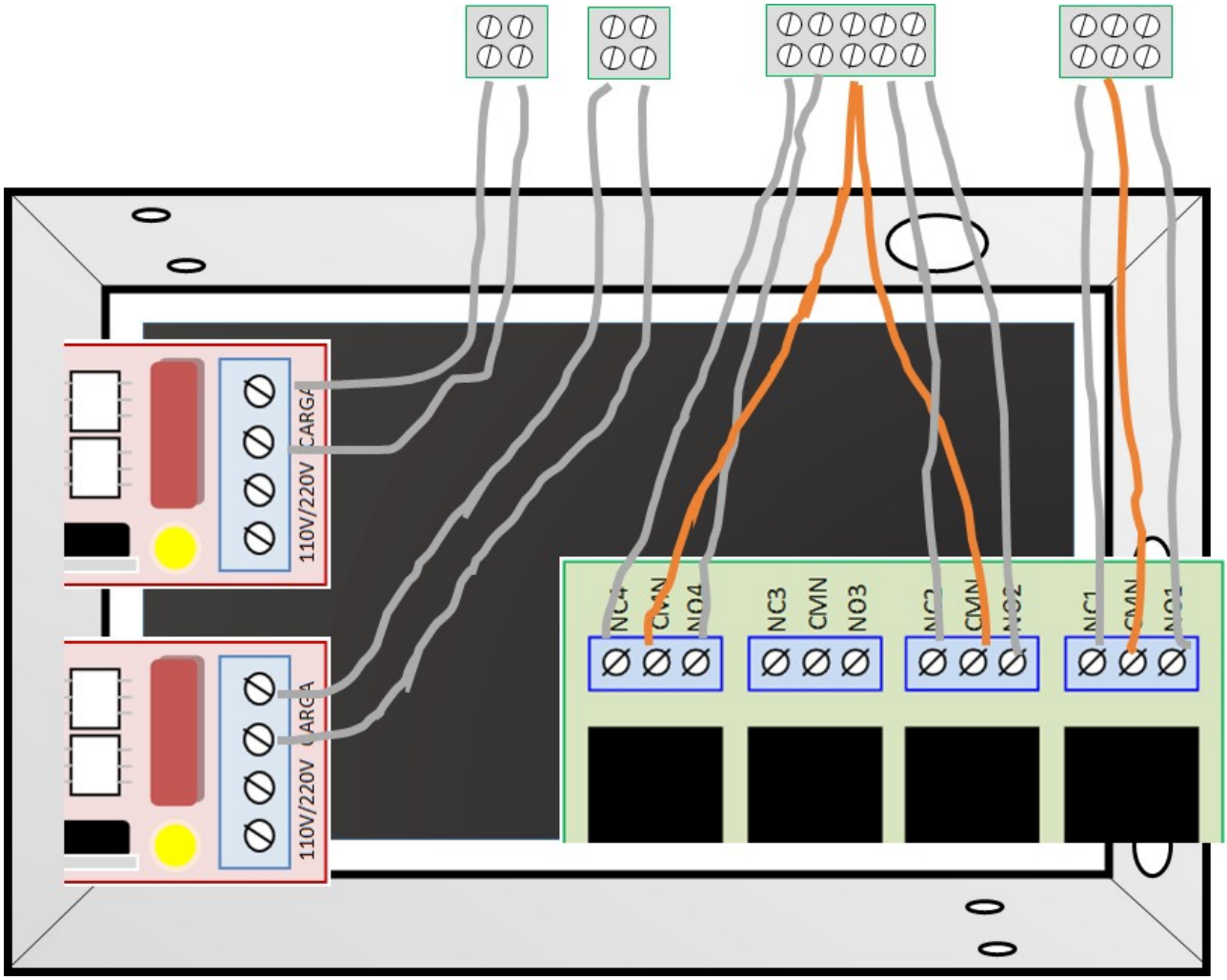
Para efetuar a ligação da *Parte 3*, basta seguir os orientações do esquema abaixo, que, como podemos ver, é da versão antiga da documentação. Repare no *Chicote "E1"*, que é exatamente a ligação que precisamos fazer para a Parte 3:



Chicote C – Chicote Principal Automation

Na parte de apresentação desse manual, vimos em detalhes a evolução do *Chicote "C"* até a incorporação de outras partes de outros chicotes; e vimos também o diagrama atualizado do *Chicote "C" REV2*.

Vamos ver agora como ligar esse chicote nas placas:



Repare que as ligações são bem diretas e:

- os traços cinzas representam os fios brancos; e
- os traços laranjas representam os fios transparentes.

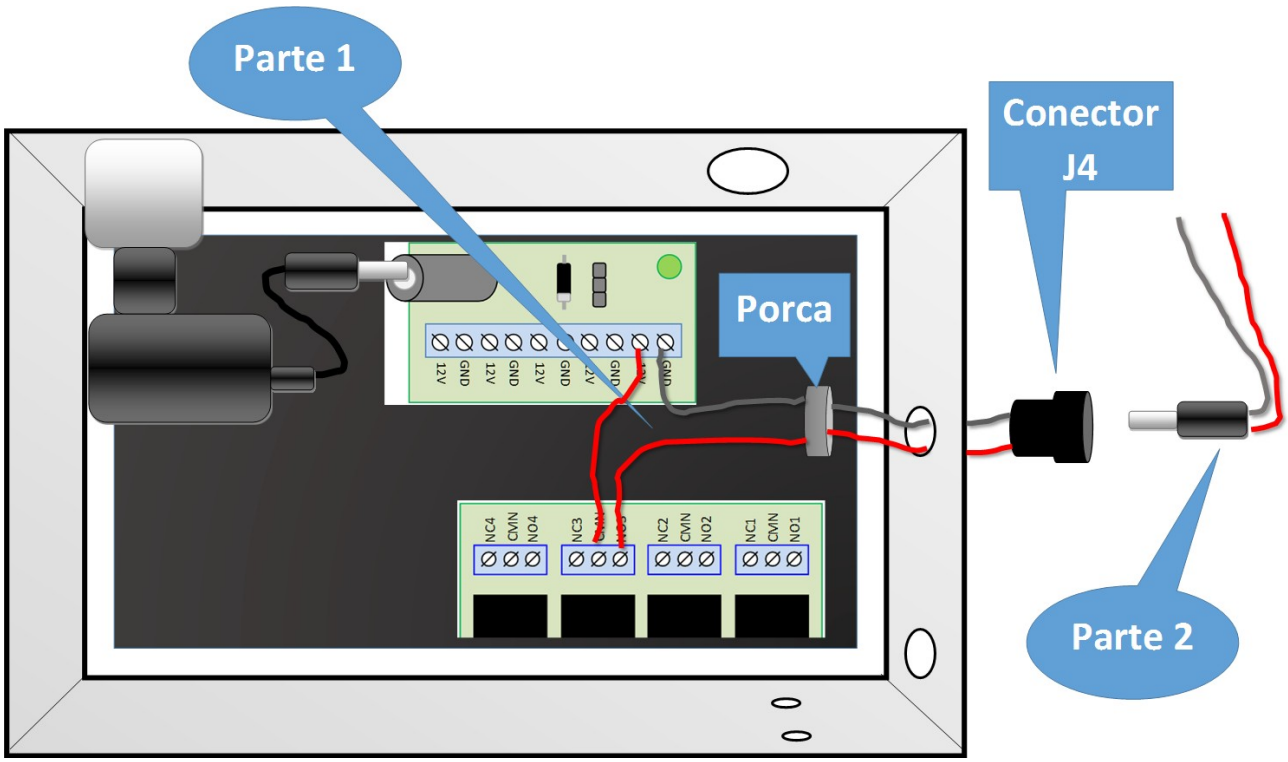
Apesar do diagrama não mostrar isso, todos esses fios são organizados por um *spagetti* e devem passar pelo *Furo "FC1"*.

Atenção – Quando efetuar as ligações, não se preocupe em saber qual fio é o da direita e qual o da esquerda, pois os fios brancos não tem polaridade entre si. Apenas ligue os fios em pares e tudo dará certo. Deixe os fios transparentes por último.

Chicote D – Chicote da Fechadura 12V Automation

Este chicote é bem fácil de ligar e, também sofreu uma pequena alteração: um *Conector Sindal* foi adicionado perto do *Conector J4* para facilitar o manuseio da porca de fixação. Antes era necessário desmontar todo *Organizador Spagetti* para manipular a porca de fixação.

Vejamos agora o esquema desse chicote:



Este *Chicote "D"* é formado por duas partes:

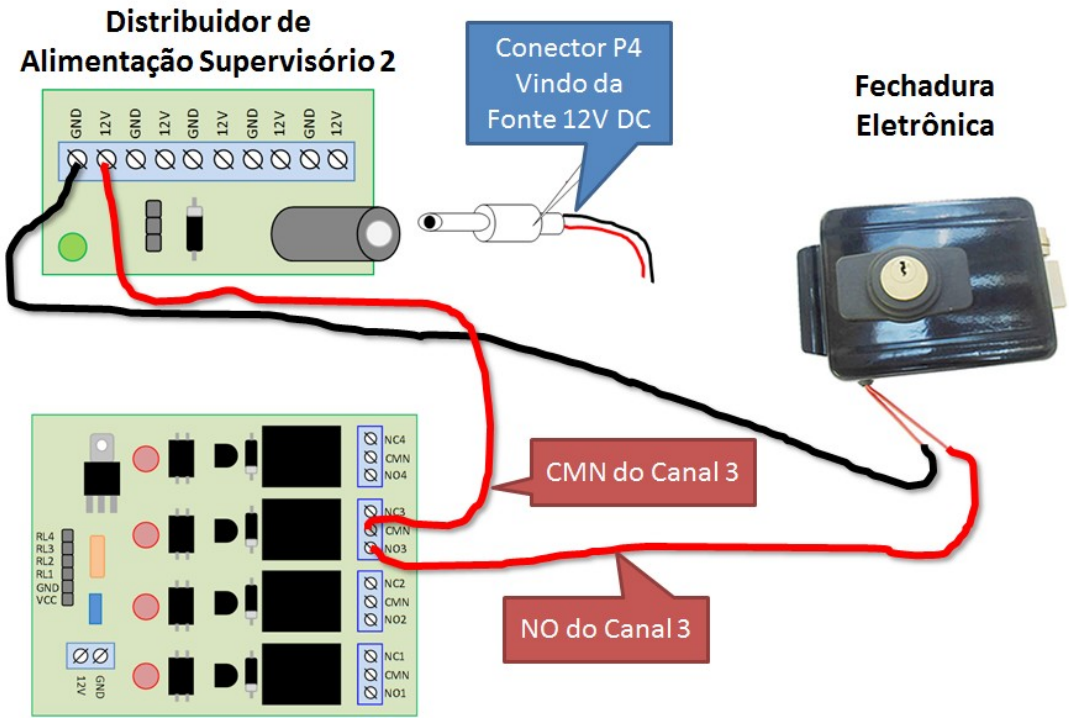
- *Parte 1* – Fiação principal, *Conector Sindal*, *Conector J4* e acabamento em *Organizador Spagetti*
- *Parte 2* – Rabicho de ligação com a *Fechadura Eletrônica* e um *Plugue P4*

A emenda da Parte 1 do chicote tem dois fios vermelhos, mas não importa a ordem de ligação, desde que sejam feitas nos terminais CMN3 e NO3, do ***Módulo de Relê 4 Canais***.

A ponta da *Parte 1* possui dois fios, um preto e um vermelho. O preto deve ser ligado no GND do ***Distribuidor de Alimentação Supervisório*** e o vermelho, no 12V.

Conforme mostrado na figura, essa fiação deve passar pelo *Furo "FD1"* e ser fixada através da porca, que acompanha o *Conector J4 Com Rosca*.

Abaixo podemos ver outro esquema mais antigo, mas que contém a *Fechadura Eletrônica*, para que possamos entender o circuito completo:



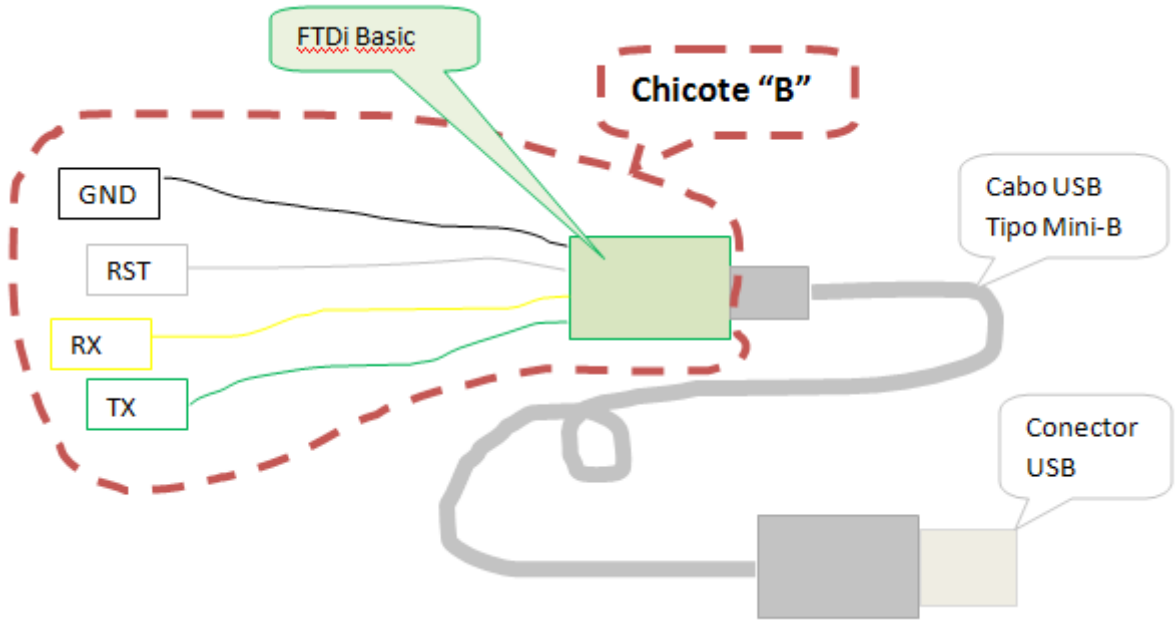
Instalação dos Programas & Drivers

Chegando nessa parte da montagem, vamos partir para instalar os programas que precisamos para que tudo funcione.

Nós temos dois (02) *Arduino Supervisórios*, no ***Kit Curso Arduino Automation***, para carregar a programação. O processo é o mesmo para ambos *Arduino Supervisórios*, exceto que vamos carregar códigos diferentes em cada *Arduino Supervisório*, mas o processo é o mesmo.

Vamos fazer um adendo aqui, pois o ***Kit Curso Arduino Automation*** será lançado já com o *Arduino Supervisório REV2*, que vamos ver que a ligação é direta!

Já para quem tiver versões antigas do *Arduino Supervisório*, precisará de um chicote que saiu de linha (consulte-nos, pois só está sendo vendido por encomenda), conforme diagrama abaixo:



Repare que este *Chicote "B"* (descontinuado) simplesmente fazia a ligação entre o *Arduino FTDi Basic* e o *Arduino Supervisório*. Agora não precisamos mais desse *Chicote*, pois o *Arduino Supervisório REV2* já vem com um conector especial para ligar o *FTDi*, conforme veremos mais adiante.

Outro problema desse chicote era que para ligar e desligar do *Arduino Supervisório*, precisávamos ficar soltando e apertando parafusos. Agora com o novo conector do *Arduino Supervisório REV2*, é por encaixe!

Um último motivo dessa melhoria que foi feita é que esse *Chicote "B"* precisava de um capacitor na linha do *Reset*, agora esse capacitor faz parte da própria placa do *Arduino Supervisório REV2*, visando aumentar a compatibilidade com outros *Conversores USB/Serial*.

Tenha em mente que o *FDTi* é conhecido por ser um *Conversor USB/Serial Universal*, ou seja, funciona em todos os *Sistemas Operacionais*, em todas as versões!

Atenção! A Serial Link está sempre pensando no público brasileiro, e não no italiano ou chinês, mas no BRASILEIRO! Portanto, tivemos a coragem de remar contra a maré, pois não concordamos com a decisão da equipe italiana do Arduino em substituir o chip FTDi pelo ATMega8u2 (ou similar). Foi uma decisão que se mostrou errada e nós da Serial Link, gostamos de pensar que somos inteligentes, e mudamos de opinião! Voltamos atrás e trabalhamos com o FTDi. Vocês não verão pela internet notícias que o Arduino FTDi Basic desgravou ou não aparece mais na lista de dispositivos USB, coisa que é comum nos Arduino UNO atuais!

Para conectar o *Arduino FTDi Basic* no *Arduino Supervisório REV2*, vide documentação do produto. Mas é só uma questão de encaixe, como podemos ver na foto abaixo:

[Foto do Encaixe do FTDi no Arduino Sup. REV2]

Instalando Drivers do FTDi Basic no Windows

Conforme vimos acima, precisamos do ***Arduino FTDi Basic*** para carregar nossos códigos no ***Arduino Supervisório***. O ***Kit Curso Arduino Automation*** contém 2x *Arduino Supervisório* e 1x *Arduino FTDi Basic* incluídos em seu pacote. O instalador precisa conectar o *FTDi Basic* no *Arduino Supervisório* e também ligar o *Cabo USB* em um computador. Nesta parte do manual, veremos as diferentes instalações do *Driver do FTDi* para as diferentes plataformas: *Windows*, *MAC OSX* e *Linux*. Todas as instruções funcionam para sistemas *32* ou *64 bits*. O *Driver do FTDi* é conhecido por não possuir restrições de funcionamento nos mais diversos *Sistemas Operacionais* (inclusive *Android*)!

Bom, antes de mais nada, independente do Sistema Operacional do seu computador, vale a pena dar uma lida no *Guia de Instalação do Arduino*, que escrevi, e até hoje é muito utilizado e continua bem atual:

http://renatoaloi.blogspot.com.br/2011/10/instalando-arduino-guia-completo.html

Observação: Nesse momento, *não* precisamos conectar o ***Arduino FTDi Basic*** no ***Arduino Supervisório***, podemos apenas conectá-lo no *computador* e deixá-lo solto na mesa, por enquanto.

Instalando Driver FTDi no Windows XP/Vista/7/8

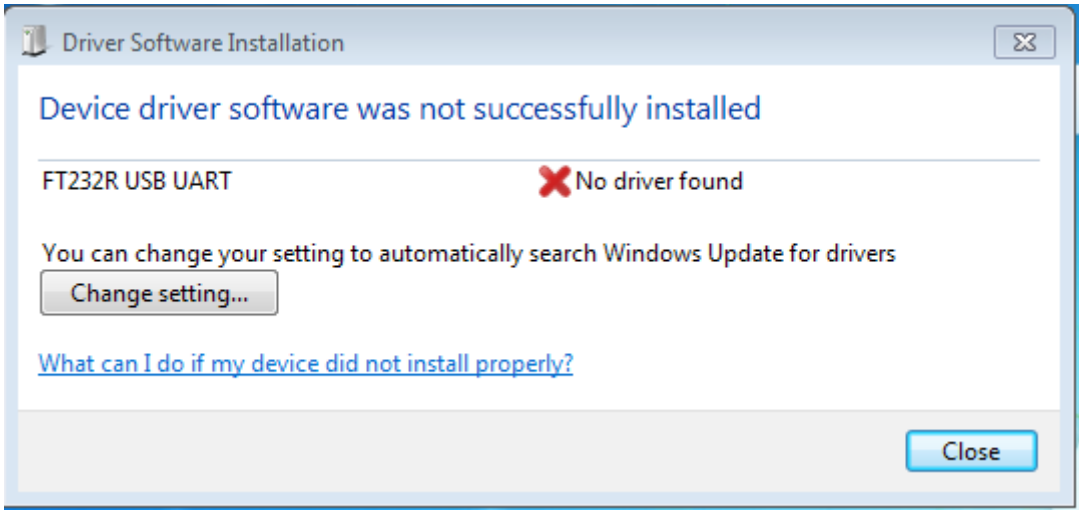
A instalação no Windows é bem simples, bastando escolher a versão certa do Driver para baixar. Acessar diretamente o site http://www.ftdichip.com , na página de Drivers, conforme segue o link:

http://www.ftdichip.com/FTDrivers.htm

Nessa página o instalador poderá encontrar *Drivers* para todas as plataformas e versões, inclusive *64 bits*!

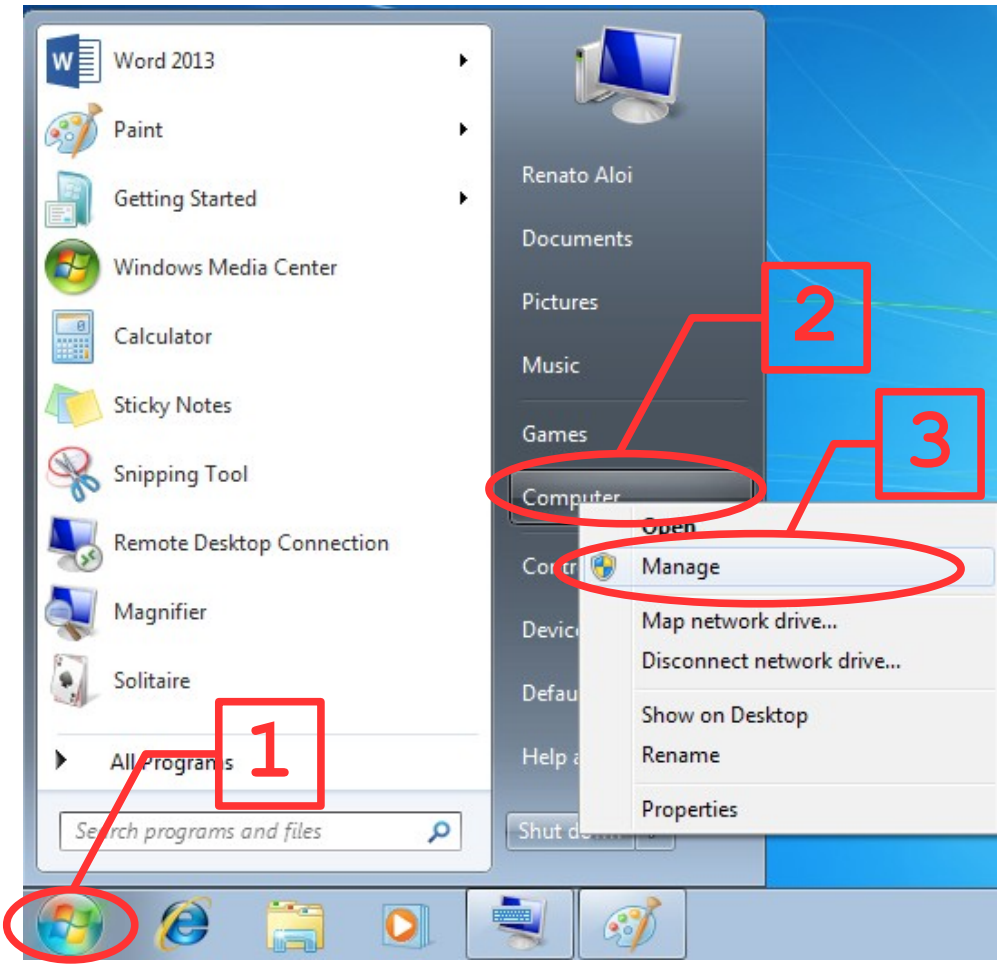
Vejamos as telas e etapas de configurações no ***Windows 7 – 32 bits***, caso seu *Windows* seja diferente, apenas selecione o arquivo diferente na hora do *Download* do *Driver*. Vamos ver isso em tempo.

Primeiro, quando conectamos nosso ***Arduino FTDi Basic*** no computador, com Windows, uma tela similar a essa abaixo aparece, informando que não foi instalado o Driver:


Driver Software Installation
Device driver software was not successfully installed
FT232R USB UART
No driver found
You can change your setting to automatically search Windows Update for drivers
Change setting...
What can I do if my device did not install properly?
Close

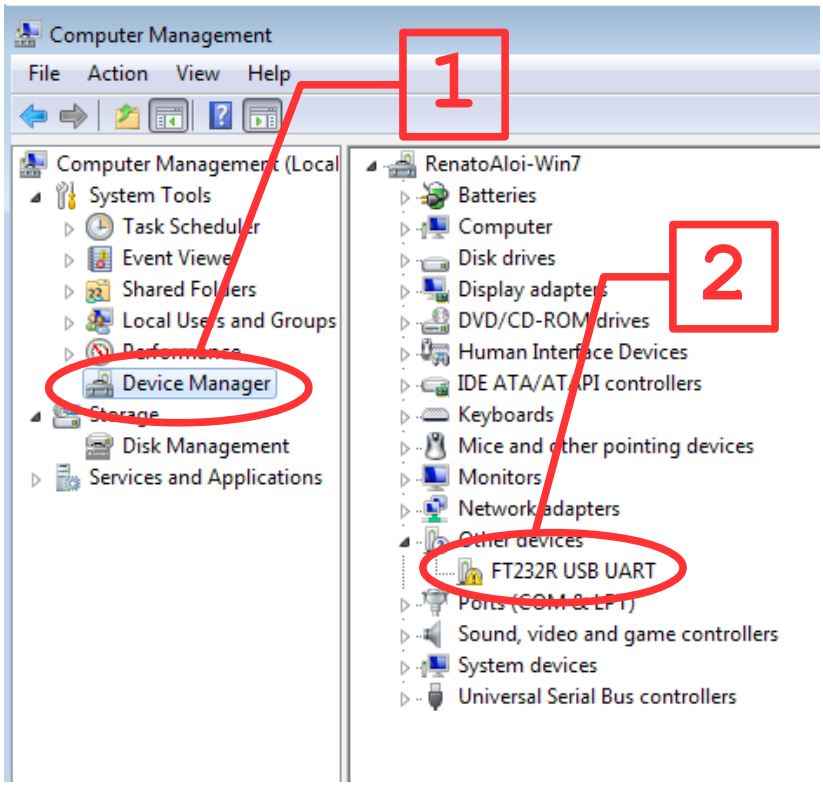
Vamos ver então uma configuração passo-a-passo do Driver do FTDi no Windows.

Passo 1 – Abrindo o Gerenciador de Dispositivos



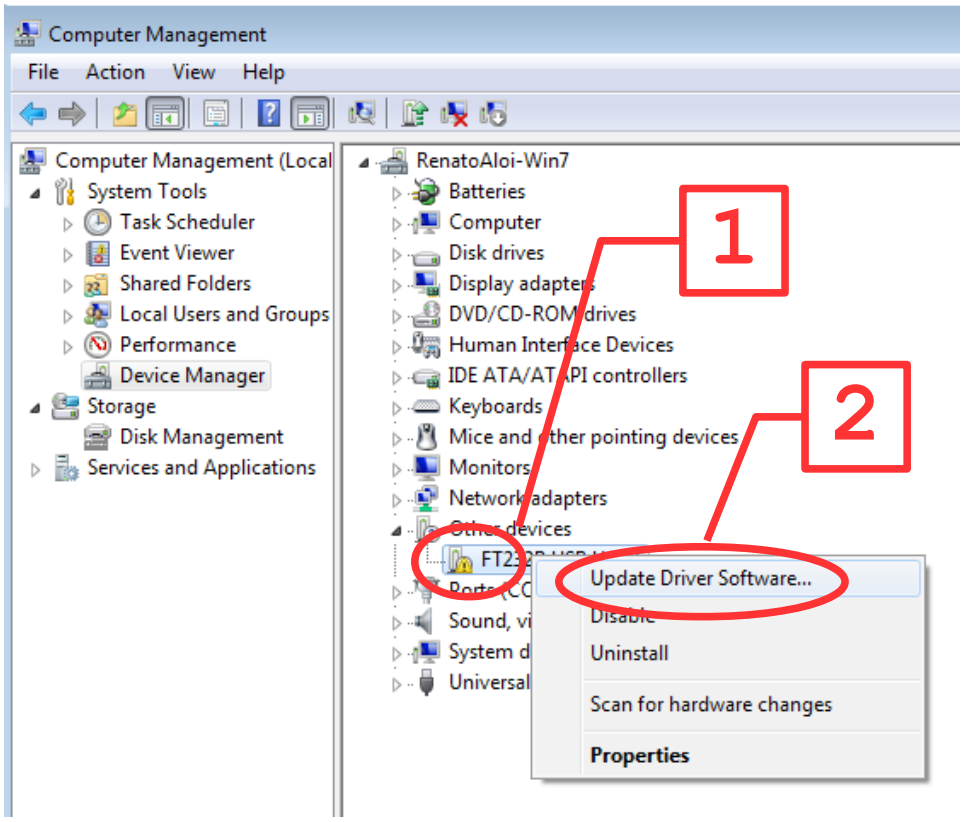
1. Clique no botão do Windows
2. Clique com o botão direito do mouse no item Computador (Computer)
3. No menu de contexto, clique na opção Gerenciar (Manage)

Passo 2 – Localizando o Dispositivo FT232RL



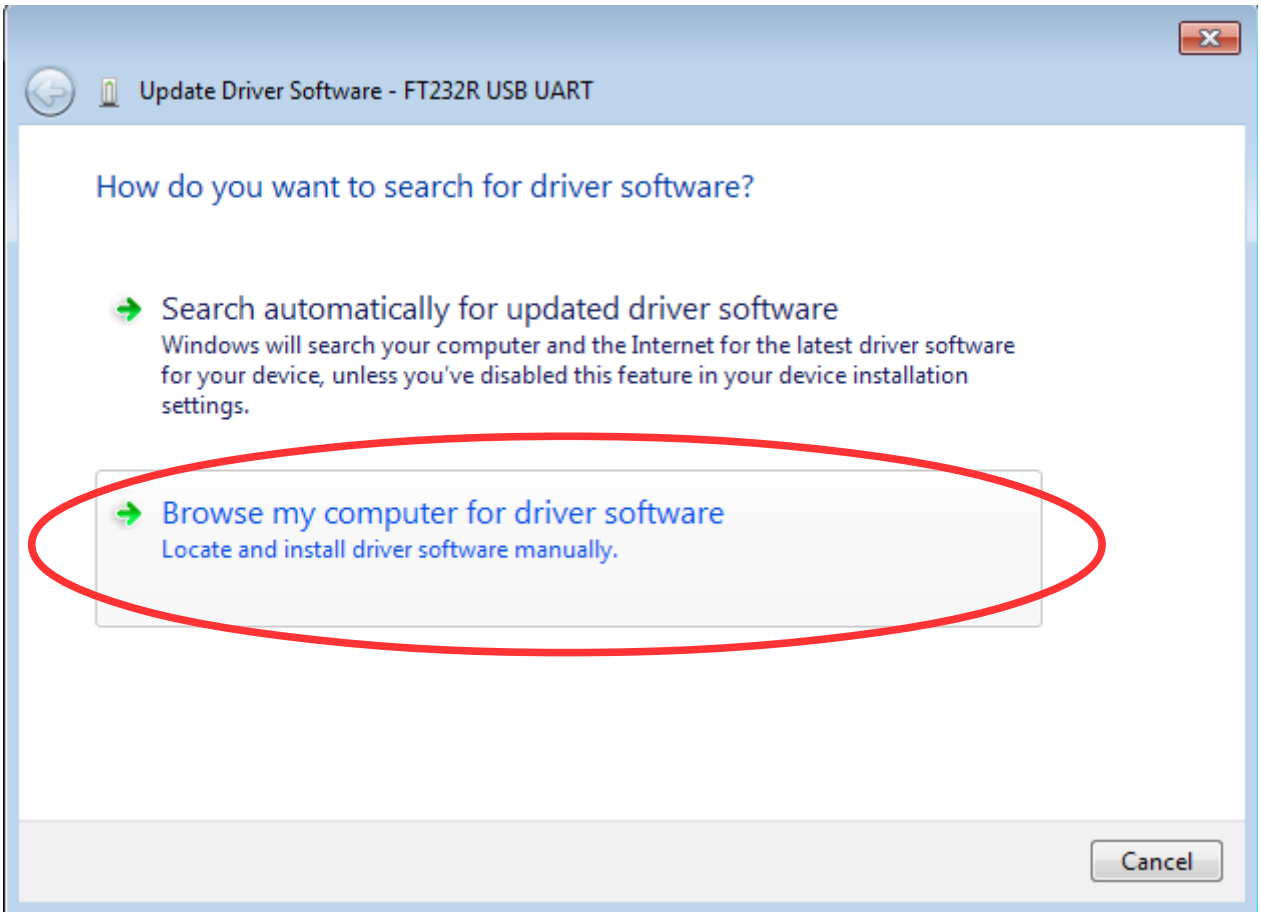
1. Clique no item Gerenciador de Dispositivos (Device Manager)
2. Localize o item FT232R USB UART, sinalizado com um triângulo amarelo com um símbolo de exclamação. Podem existir outros itens assim, mas o que nos interessa é esse.

Passo 3 – Atualizando Driver no Computador



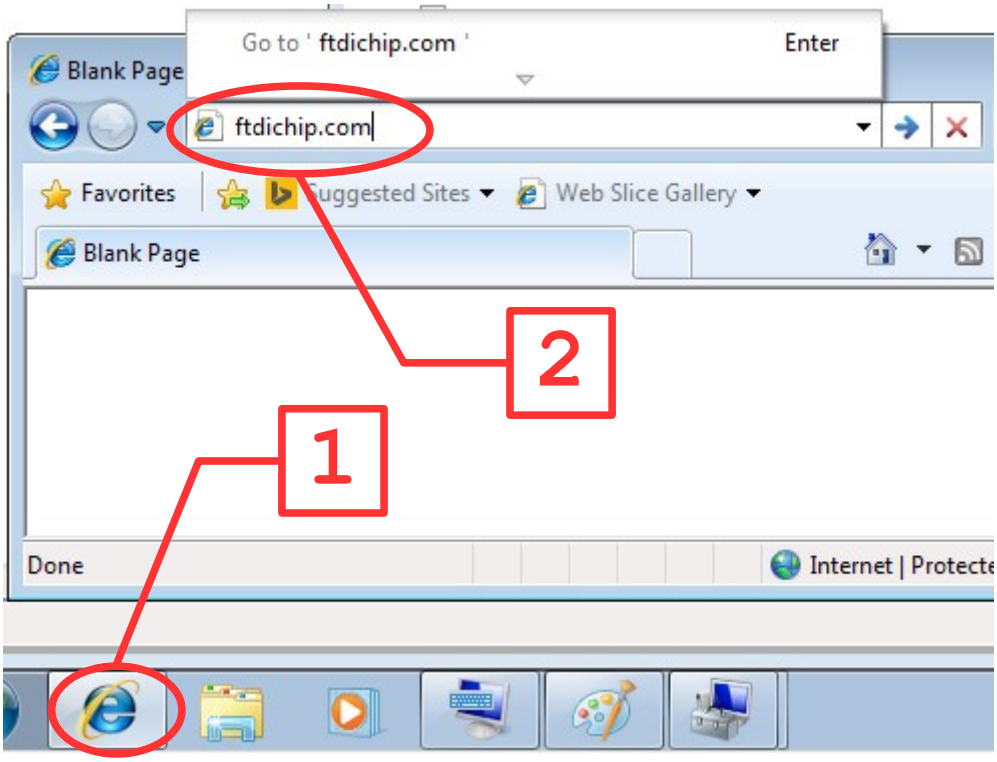
1. Clique com o botão direito no item FT232RL USB UART para exibir o menu de contexto
2. Clique na opção Atualizar Driver... (Update Driver Software...)

Passo 4 – Procurando o Driver no Computador


Update Driver Software - FT232R USB UART
How do you want to search for driver software?
Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.
Browse my computer for driver software
Locate and install driver software manually.
Cancel

Clique na opção mostrada acima, em português é algo como: “Procurar no computador por drivers”, basicamente é a segunda opção.

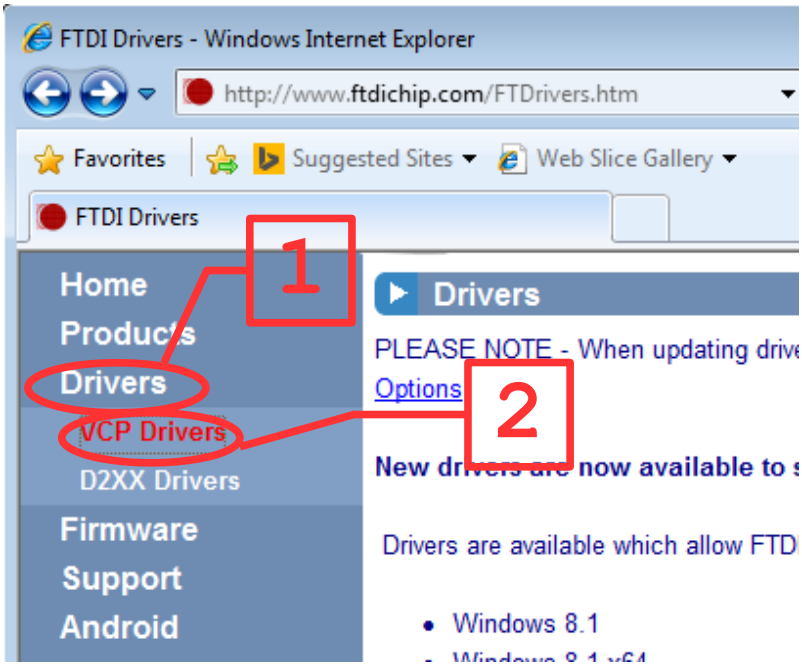
Passo 5 – Baixando o Driver



Deixe a tela de atualização do Driver aberta e execute as tarefas abaixo:

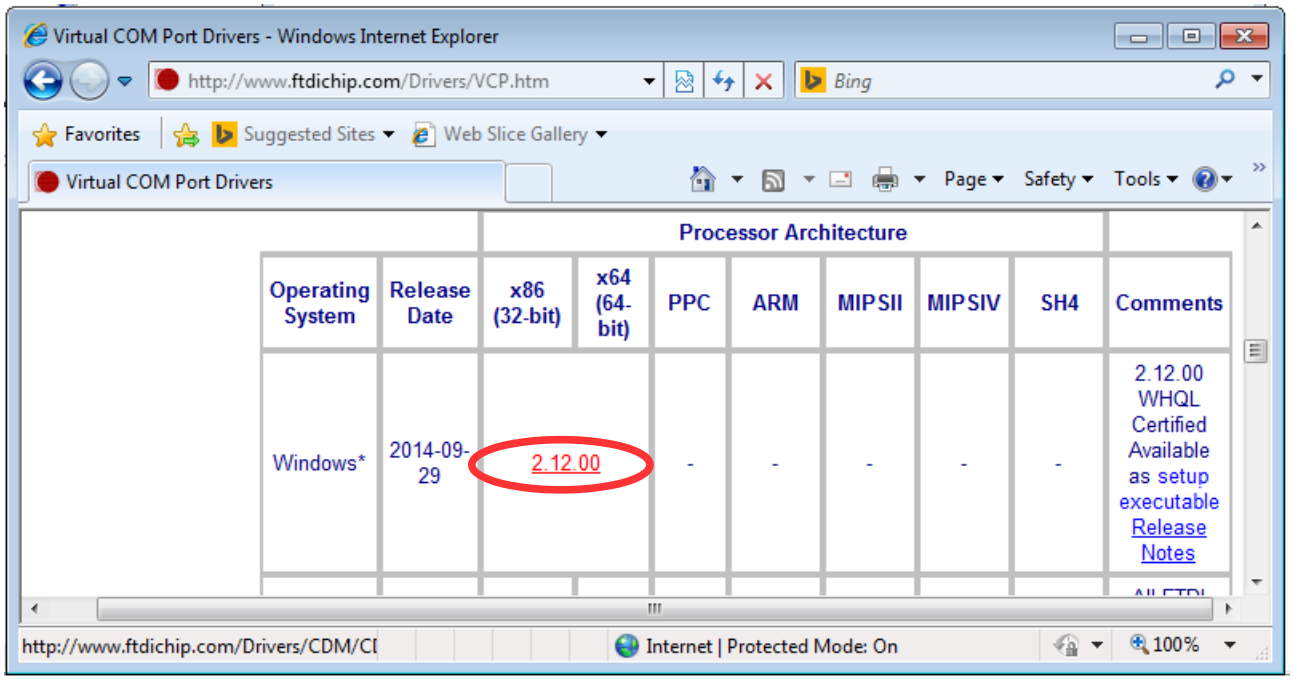
1. Clique no ícone do Internet Explorer ou utilize o Navegador de sua preferência
2. Digite ou acesse o link http://www.ftdichip.com

Passo 6 – Acessando a Página de Download de VCP Drivers



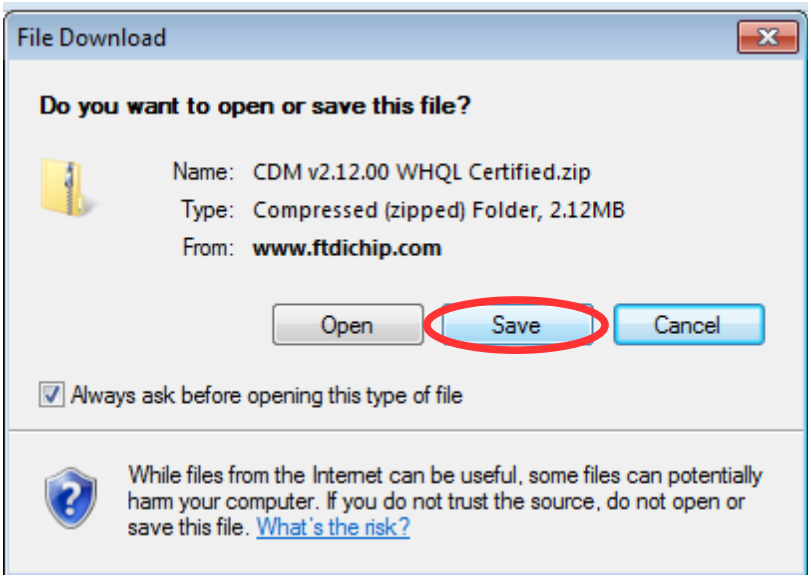
1. No menu esquerdo, clique na opção “Drivers”
2. No sub-menu que abrir, clique no item “VCP Drivers”

Passo 7 – Selecionando Opção de Download



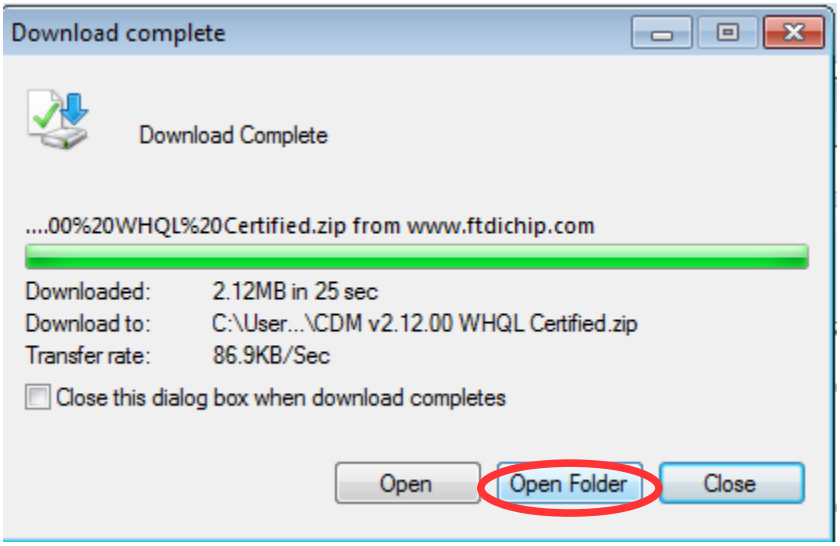
Primeiro role a tela para chegar nessa tabela, mostrada na imagem acima. Depois clique no link indicado na imagem que serve tanto para todas versões de Windows. Até para os 64-bits.

Passo 8 – Salvando o Download


File Download
Do you want to open or save this file?
Name: CDM v2.12.00 WHQL Certified.zip
Type: Compressed (zipped) Folder, 2.12MB
From: www.ftdichip.com
Open
Save
Cancel
Always ask before opening this type of file
While files from the Internet can be useful, some files can potentially harm your computer. If you do not trust the source, do not open or save this file. What's the risk?

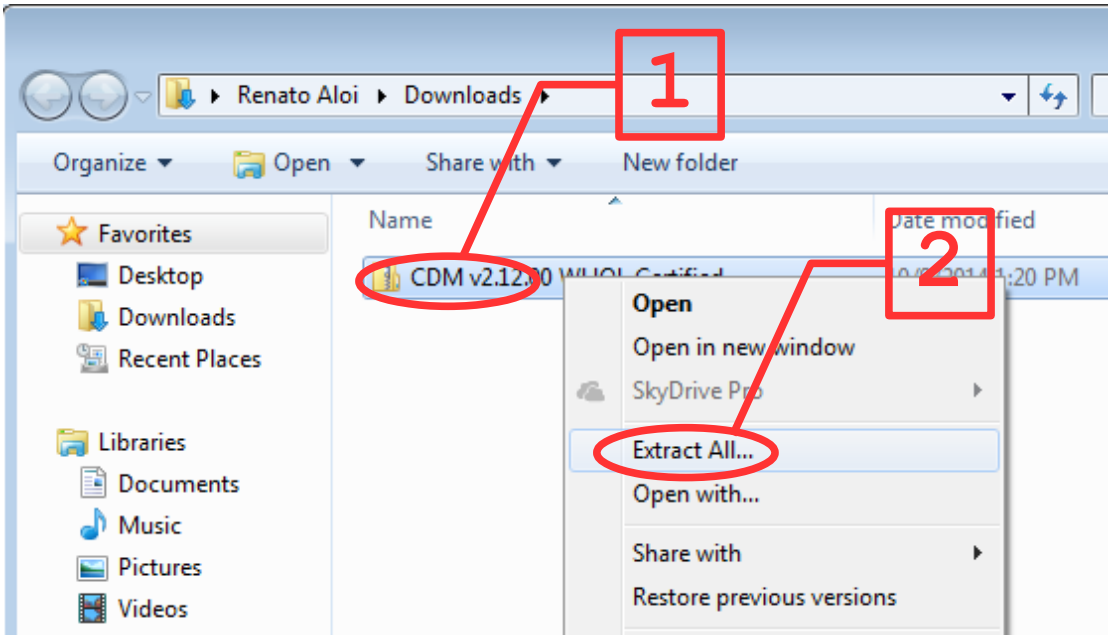
Na tela que aparecer, clique no botão Salvar (Save), conforme mostrado acima.

Passo 9 – Abrindo Pasta do Download



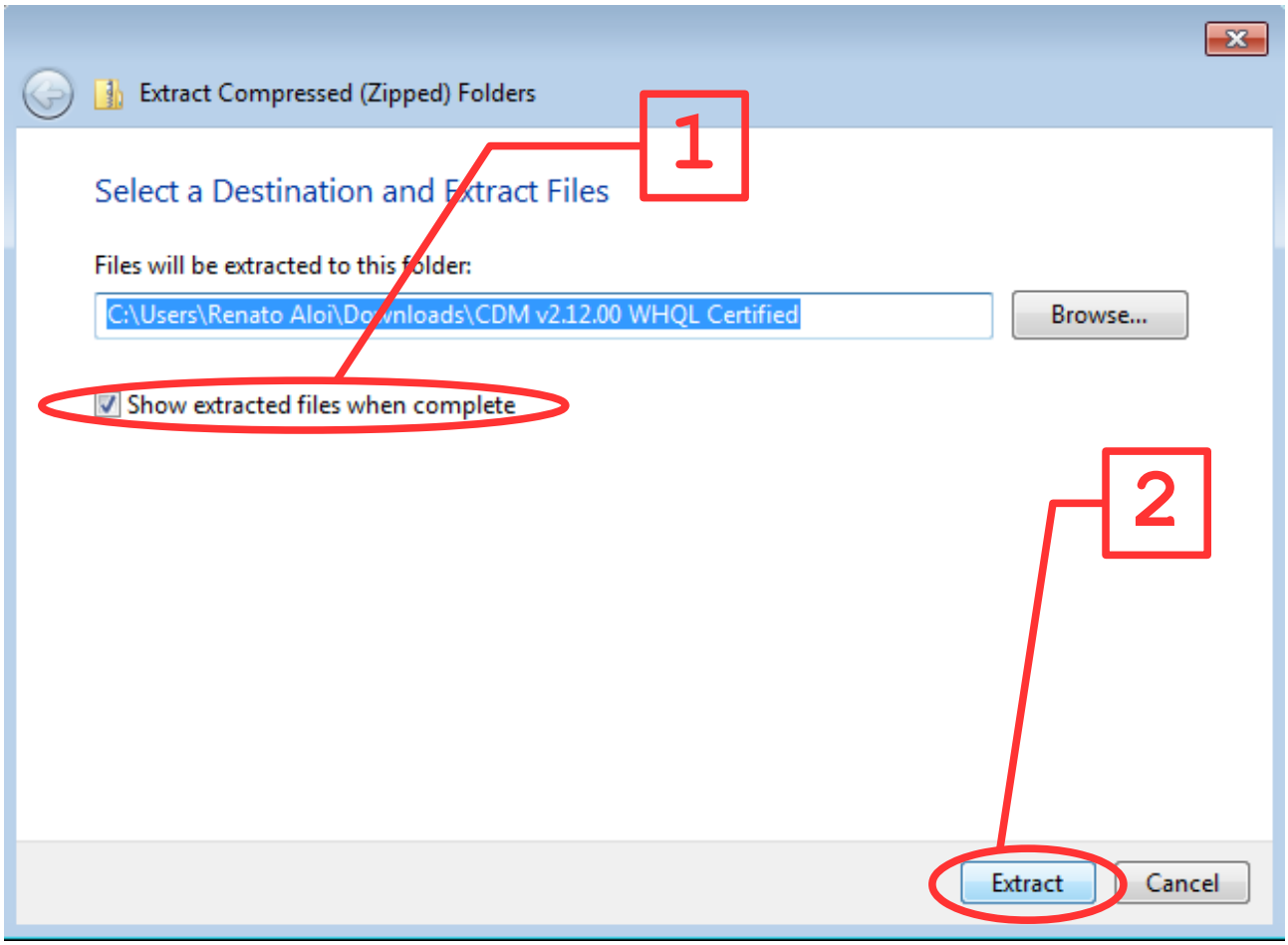
Clique no botão Abrir Pasta (Open Folder), conforme mostrado acima.

Passo 10 – Extraindo o Pacote



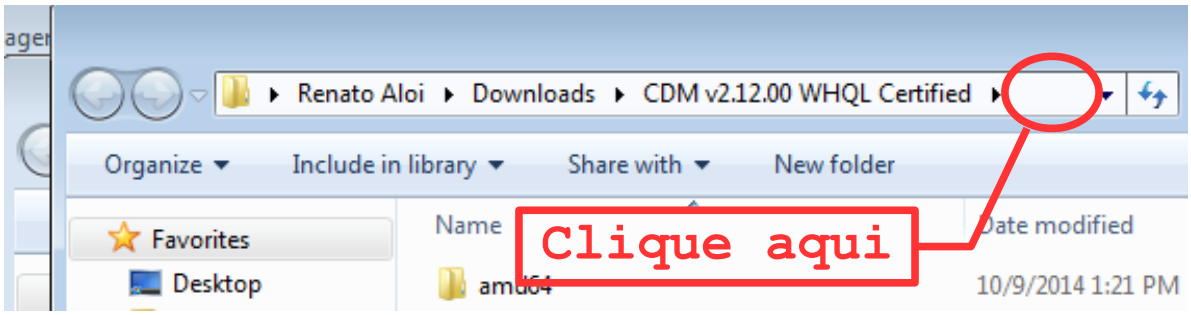
1. Clique com o botão direito do mouse no pacote compactado, recentemente baixado.
2. No menu de contexto que aparecer, clique em “Extrair Tudo...” (Extract All...)

Passo 11 – Escolha a Pasta de Destino da Descompactação



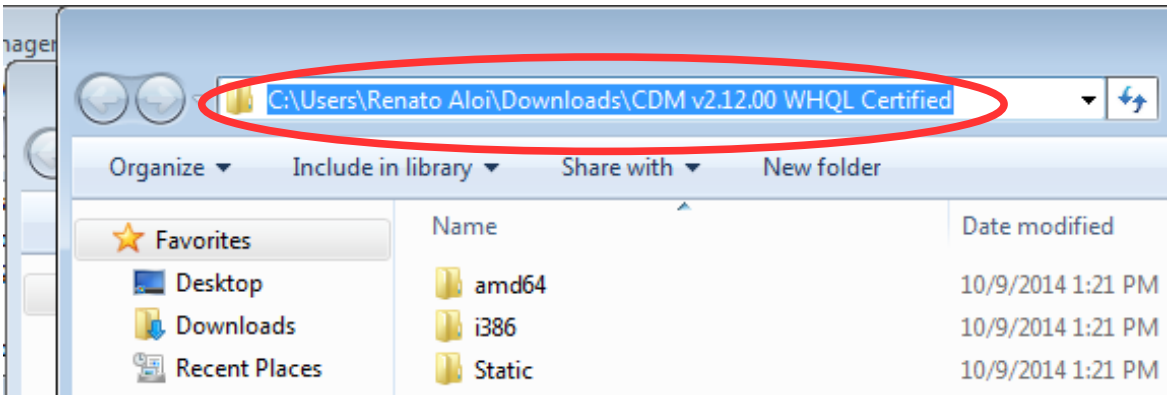
1. Mantenha a opção “Mostrar arquivos extraídos” (Show extracted files when complete) ativada
2. Clique no botão Extrair (Extract)

Passo 12 – Pegando o Caminho da Pasta (Parte 1)



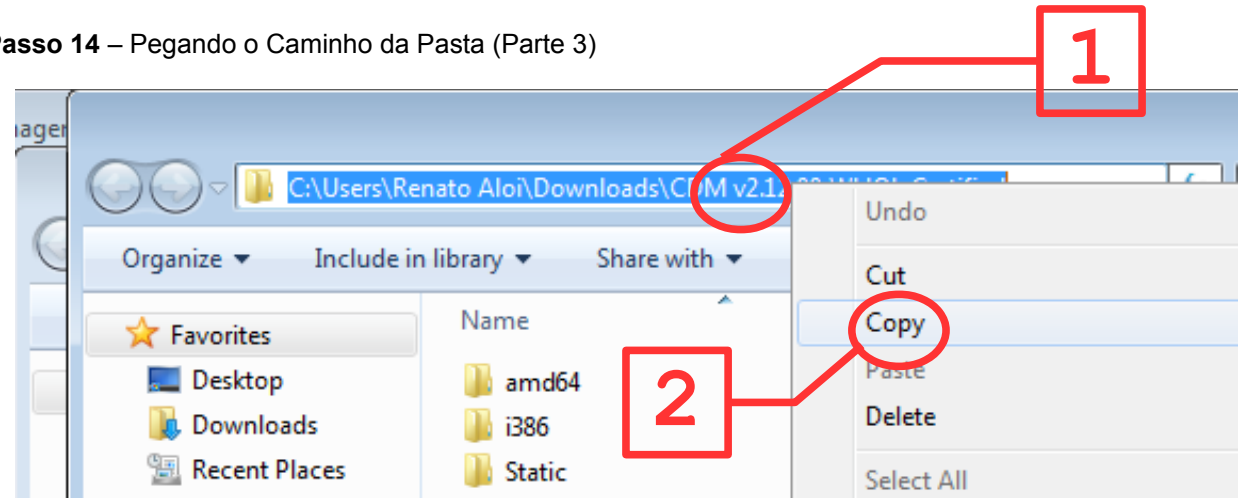
Na pasta que os arquivos foram descompactados, dê um clique normal com o mouse na área indicada na figura acima. E repare no passo seguinte o que acontece.

Passo 13 – Pegando o Caminho da Pasta (Parte 2)



Repare que ao clicar na área do Passo 12, o caminho fica no formato "texto" e podemos copiar agora. Primeiro selecione todo o caminho, como mostrado na imagem acima.

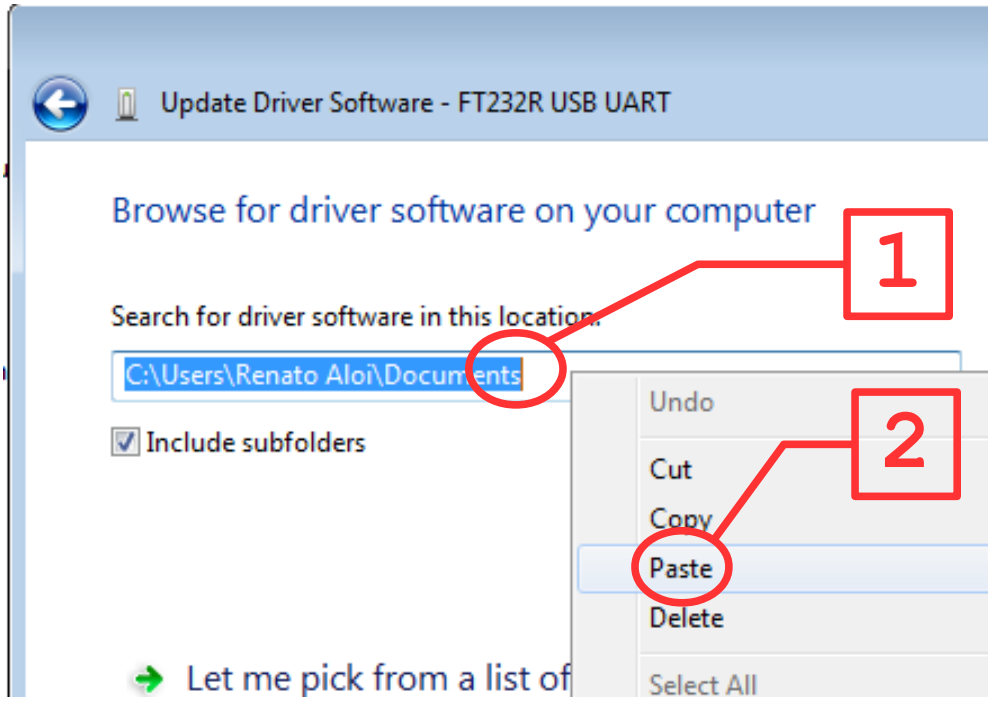
Passo 14 – Pegando o Caminho da Pasta (Parte 3)



Finalmente, com todo o caminho da pasta selecionado, siga os itens abaixo:

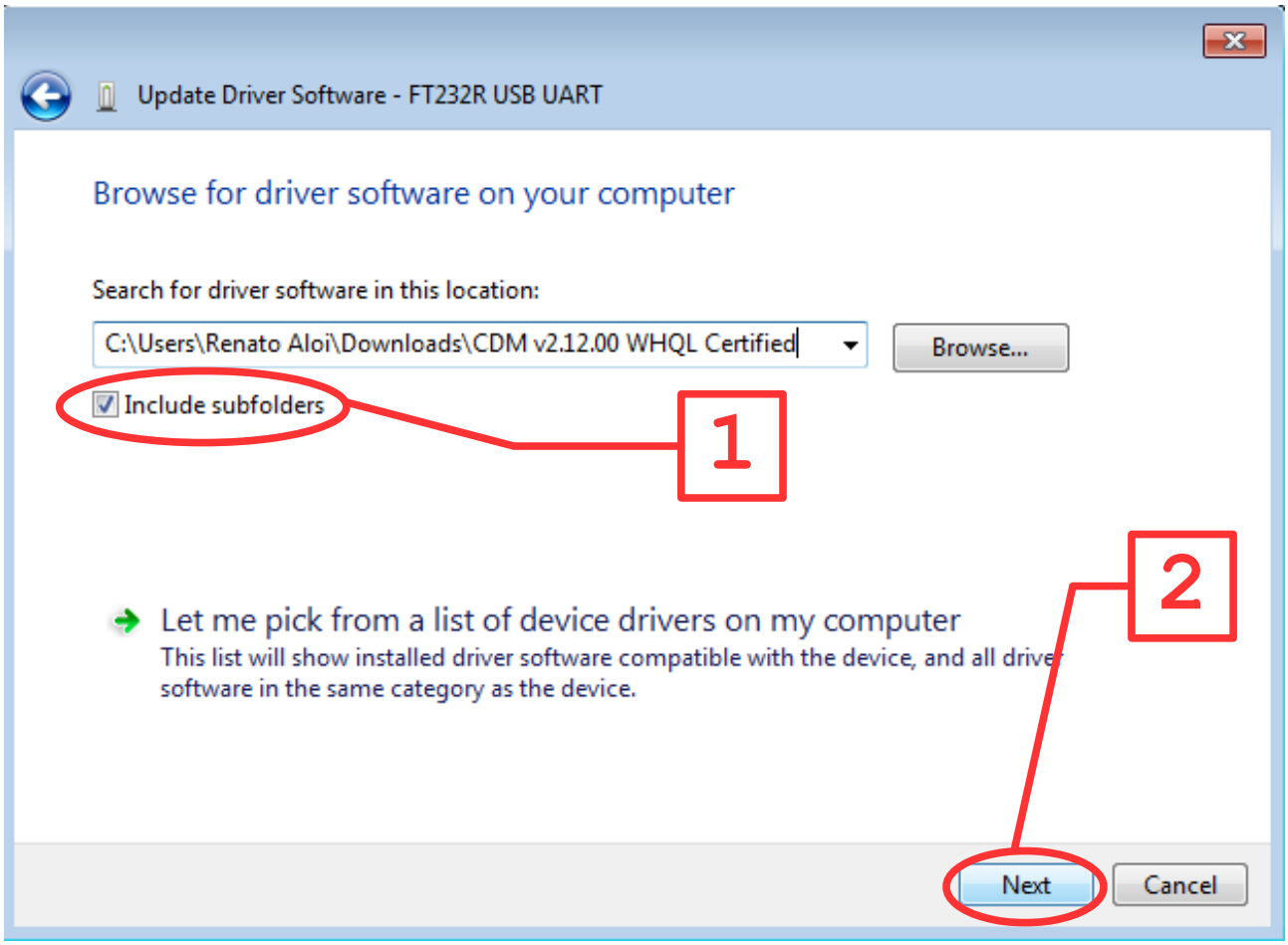
1. Clique com o botão direito do mouse em qualquer parte selecionada;
2. Clique na opção Copiar (Copy)

Passo 15 – Colando Caminho da Pasta Na Pesquisa de Drivers



1. Clique com o botão direito do mouse em qualquer parte selecionada;
2. Clique na opção Colar (Paste)

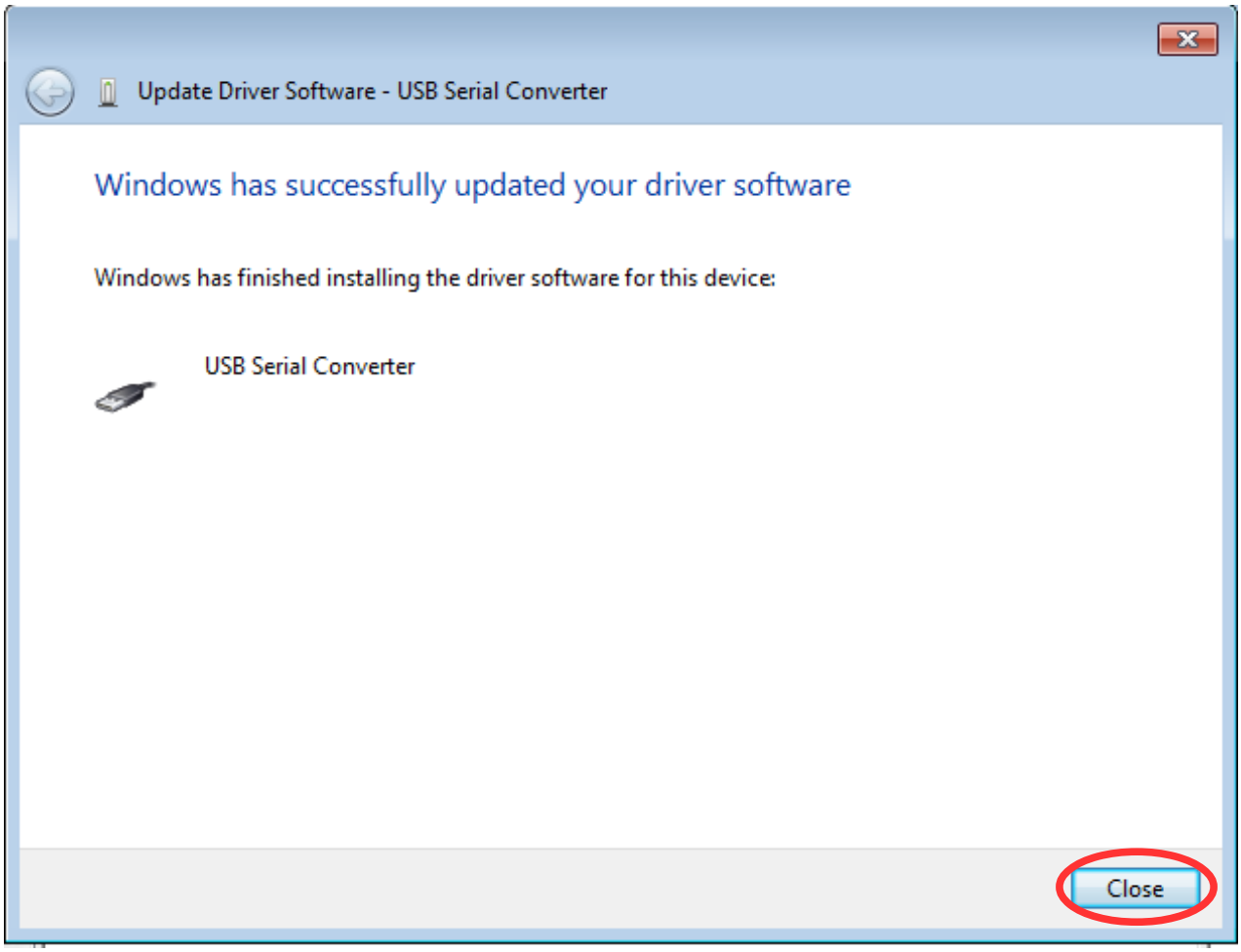
Passo 16 – Verificando Sub-Pastas & Continuando


Update Driver Software - FT232R USB UART
Browse for driver software on your computer
Search for driver software in this location:
C:\Users\Renato Aloi\Downloads\CDM v2.12.00 WHQL Certified
Browse...
Include subfolders
1
Let me pick from a list of device drivers on my computer
This list will show installed driver software compatible with the device, and all drive
software in the same category as the device.
2
Next
Cancel

Com o caminho da pasta descompactada colado no campo de pesquisa, siga os itens:

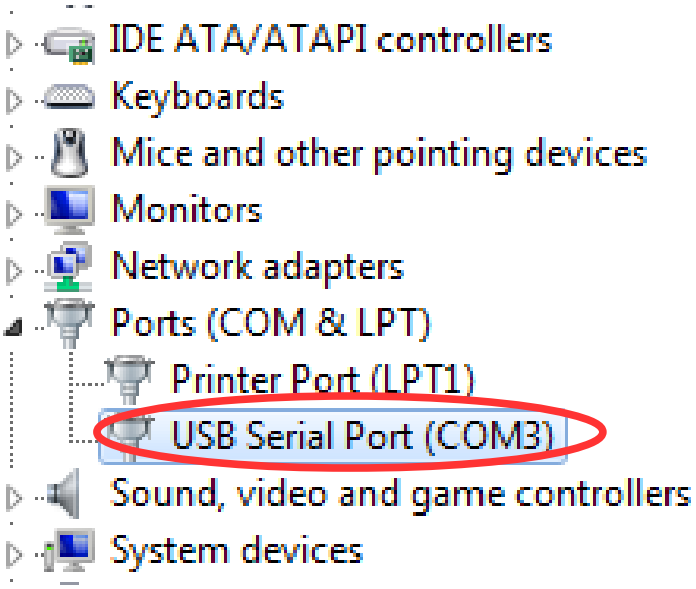
1. Selecione a opção “Incluir Sub-Pastas” (Include subfolders)
2. Clique no botão Próximo (Next)

Passo 17 – Terminando a Instalação do Driver


Update Driver Software - USB Serial Converter
Windows has successfully updated your driver software
Windows has finished installing the driver software for this device:
USB Serial Converter
Close

Pronto, quando essa tela acima aparecer, significa que o Driver foi corretamente instalado. Clique em Fechar (Close), para terminar.

Passo 18 – Verificando a Instalação do Driver



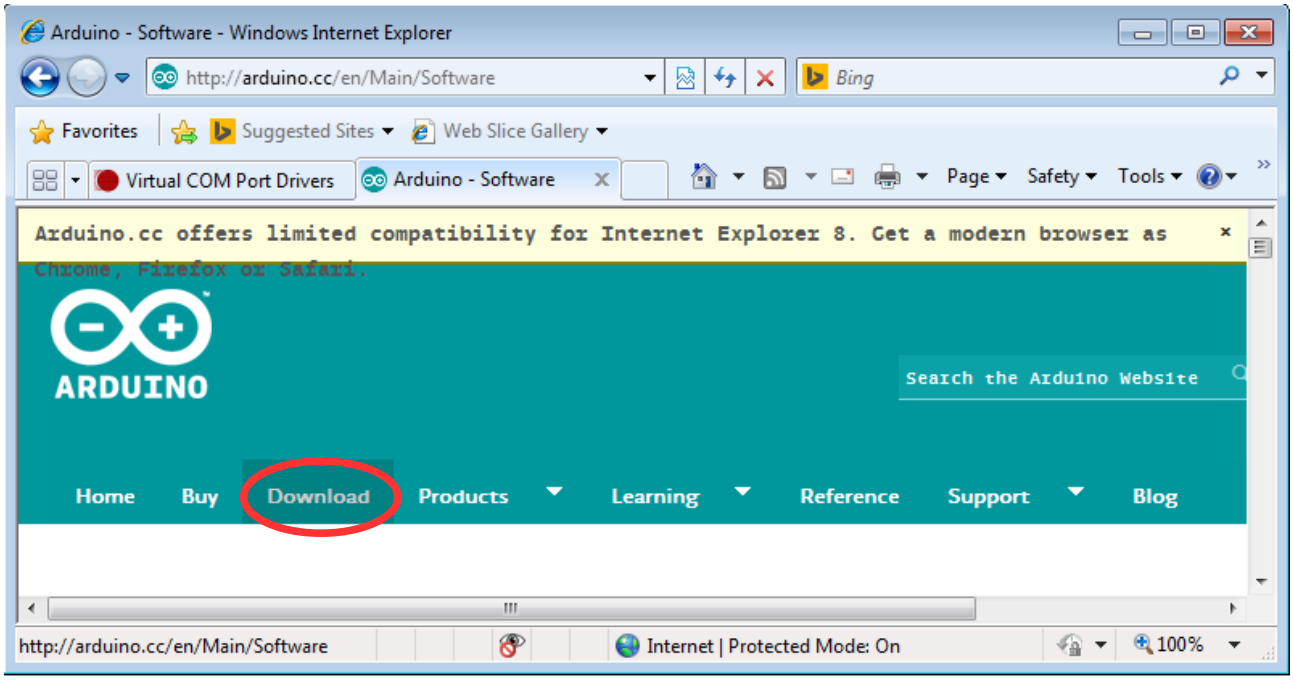
Voltando ao Gerenciador de Dispositivos (Device Manager), podemos verificar agora que uma nova porta Serial foi criada. Memorize o número da COM, no caso do nosso exemplo é COM3.

Instalando o Ambiente do Arduino no Windows

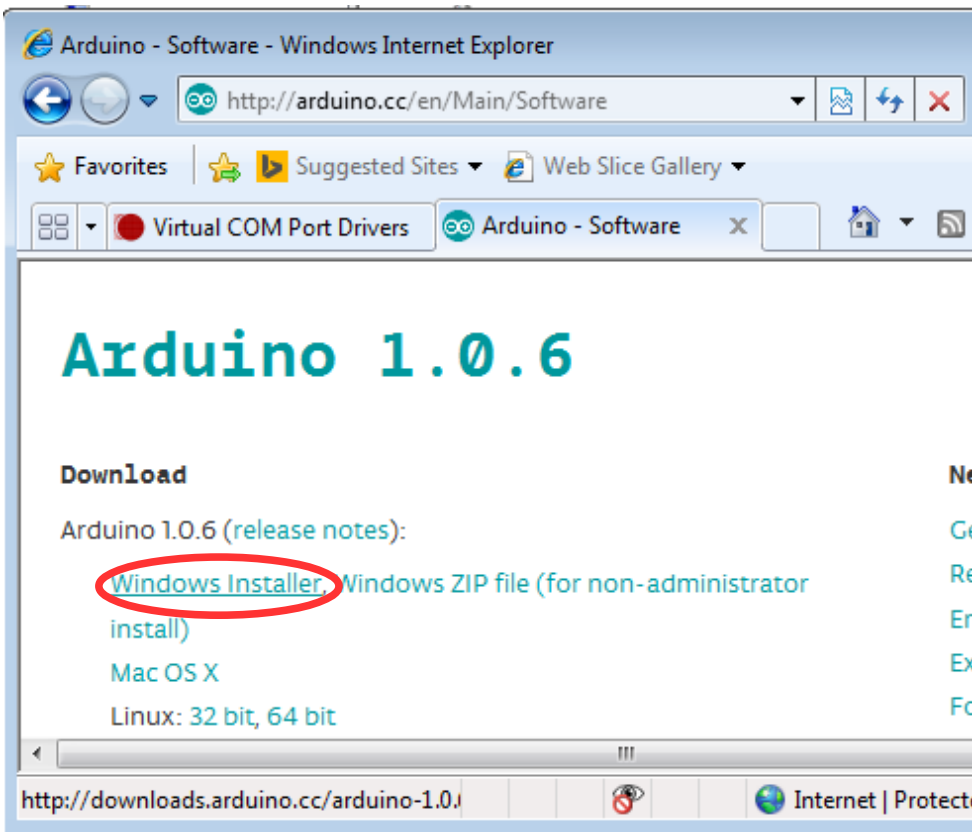
Com o ***Driver do FTDi*** corretamente instalado, precisamos de mais uma ferramenta: o *Ambiente Integrado de Desenvolvimento do Arduino*, conhecido apenas por ***Arduino IDE***.

Para a plataforma Windows, existe um instalador que facilita bastante essa parte. Vamos ao passo-a-passo.

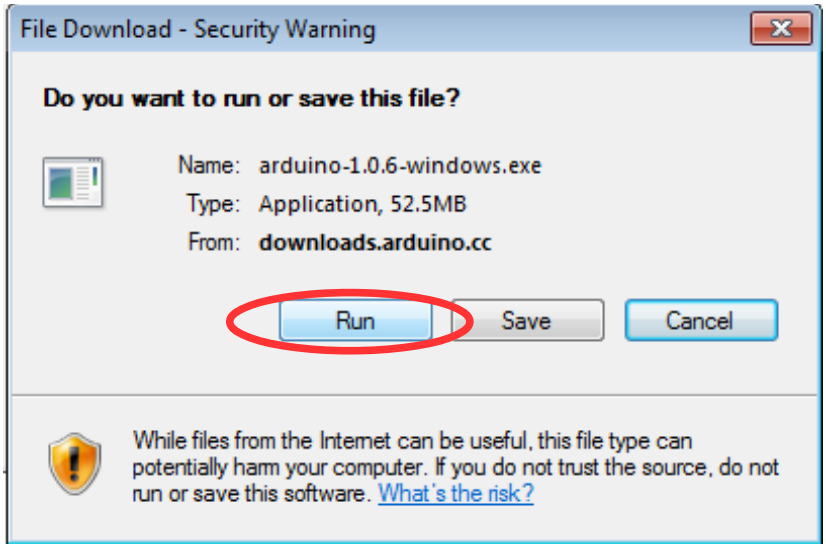
Passo 1 - Primeiro volte ao Navegador de Internet e vá até a página http://arduino.cc , clique na aba "Download", conforme mostrado abaixo:



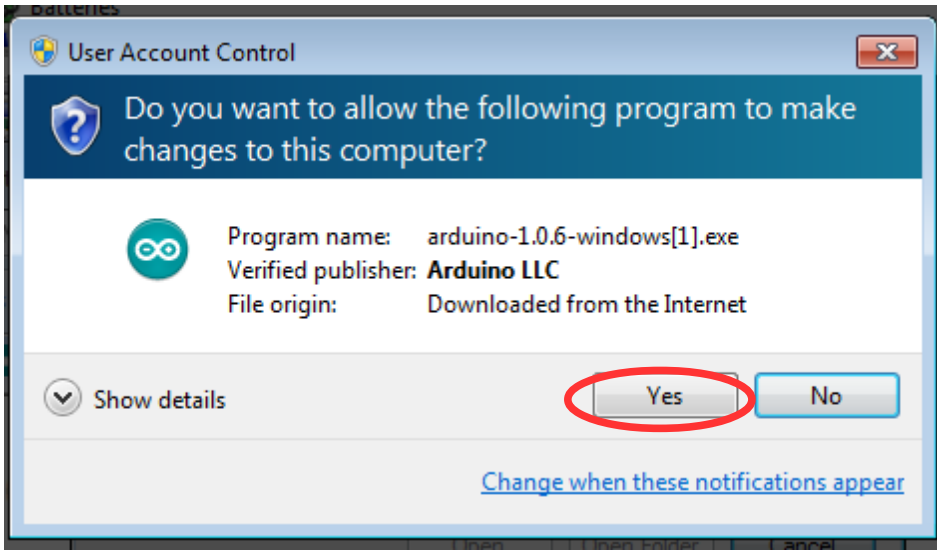
Passo 2 – Role a página até chegar na parte mostrada na imagem abaixo. Depois clique na opção "Windows Installer".



Passo 3 – Quando aparecer a janela abaixo, clique na opção que Executar (Run), pois como é um instalador, basta executar, não precisamos salvar.


File Download - Security Warning
Do you want to run or save this file?
Name: arduino-1.0.6-windows.exe
Type: Application, 52.5MB
From: downloads.arduino.cc
Run
Save
Cancel
While files from the Internet can be useful, this file type can potentially harm your computer. If you do not trust the source, do not run or save this software. What's the risk?

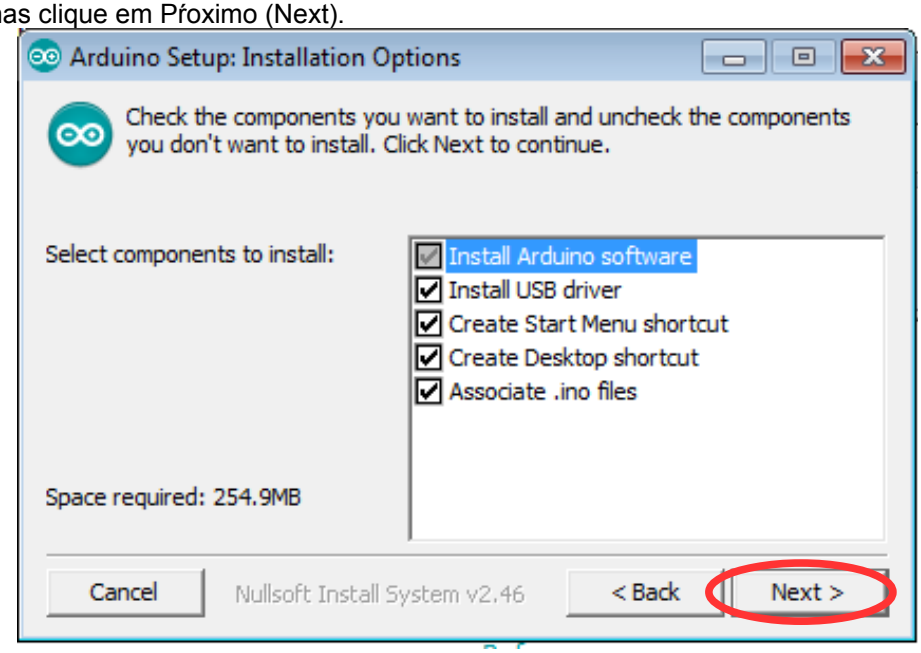
Passo 4 – Quando o site terminar o download uma imagem de autorização, como mostrada abaixo, aparecerá e solicitará sua confirmação. Se o instalador estiver usando um Windows XP, essa etapa não será necessária. Clique em Sim (Yes) para continuar.


User Account Control
Do you want to allow the following program to make changes to this computer?
Program name: arduino-1.0.6-windows[1].exe
Verified publisher: Arduino LLC
File origin: Downloaded from the Internet
Show details
Yes
No
Change when these notifications appear

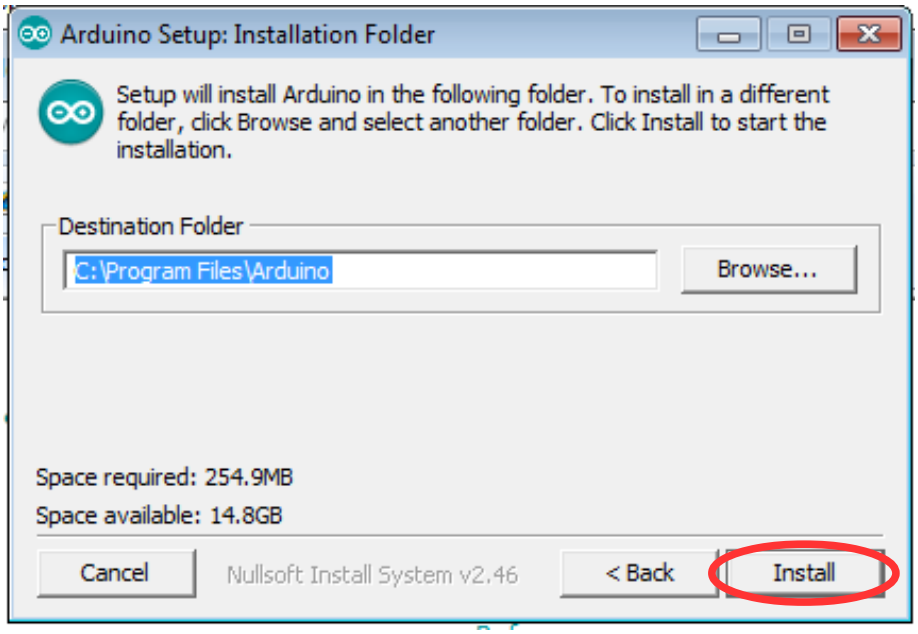
Passo 5 – A primeira tela do instalador se parece com a imagem que vemos abaixo. Leia a licensa de uso e clique em “Aceito” (I Agree), conforme marcado na imagem.

Arduino Setup: License Agreement
Please review the license agreement before installing Arduino. If you accept all terms of the agreement, click I Agree.
GNU LESSER GENERAL PUBLIC LICENSE
Version 3, 29 June 2007
Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.
Cancel
Nullsoft Install System v2.46
I Agree

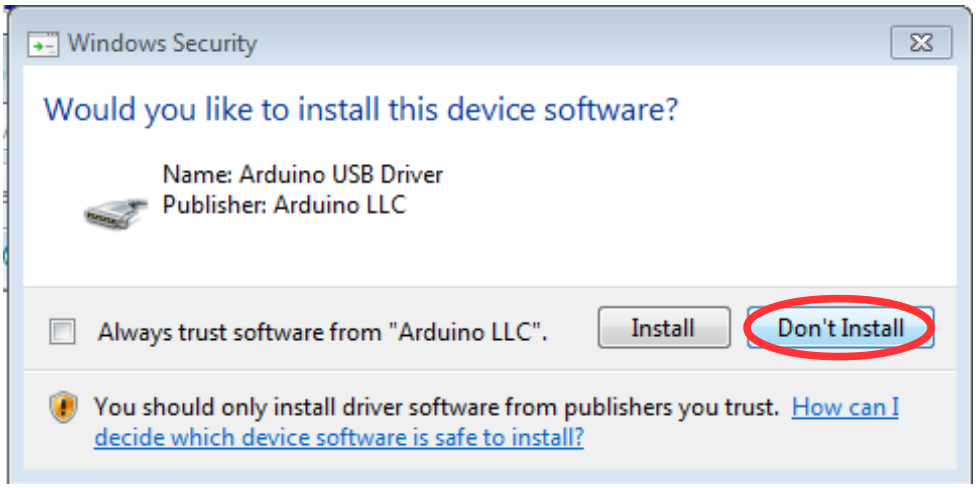
Passo 6 – Na segunda tela, temos as opções de escolher os itens a serem instalados. Não precisa mexer em nada, apenas clique em Próximo (Next).


Arduino Setup: Installation Options
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.
Select components to install:
Install Arduino software
Install USB driver
Create Start Menu shortcut
Create Desktop shortcut
Associate .ino files
Space required: 254.9MB
Cancel
Nullsoft Install System v2.46
< Back
Next >

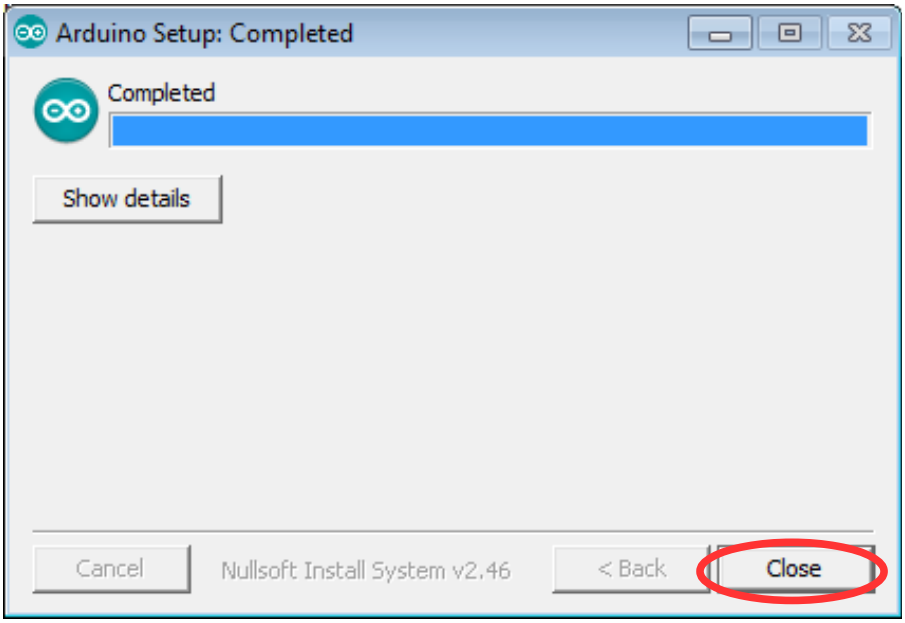
Passo 7 – Nesta tela o instalador pode escolher o diretório onde o Arduino será instalado. Vamos deixar o padrão e clicar em Instalar (Install).


Arduino Setup: Installation Folder
Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.
Destination Folder
C:\Program Files\Arduino
Browse...
Space required: 254.9MB
Space available: 14.8GB
Cancel
Nullsoft Install System v2.46
< Back
Install

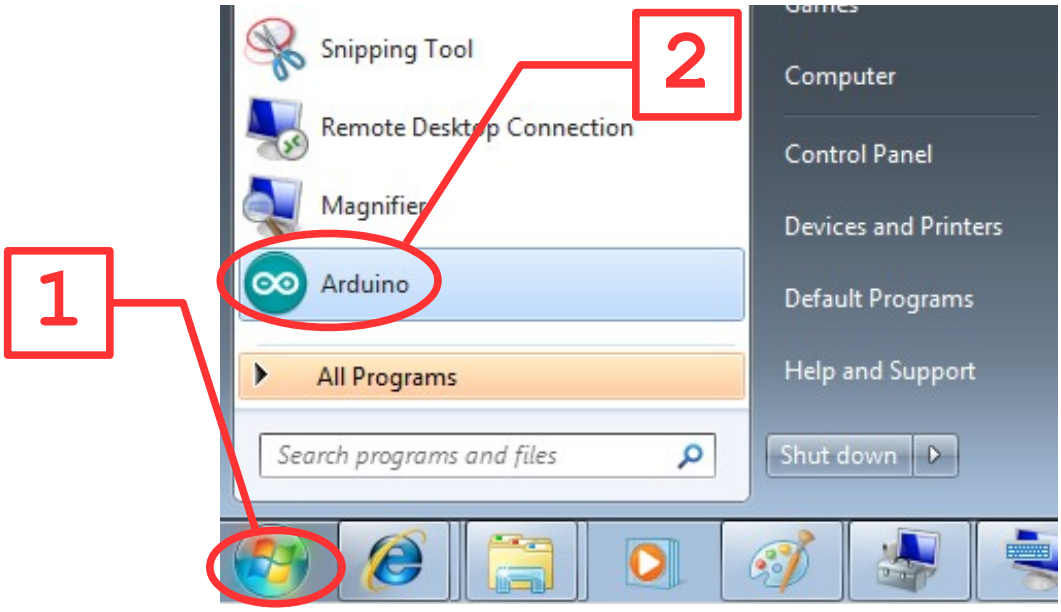
Passo 8 – Quando a instalação principal terminar, aparecerá a tela abaixo, solicitando saber se o Driver deve ser instalado. Como já fizemos isso, podemos clicar em Não Instalar (Don't Install). O instalador pode escolher clicar em Instalar (Install), caso seja utilizado futuramente um Arduino UNO, no nosso caso vamos clicar em Não, mesmo, conforme segue.


Windows Security
Would you like to install this device software?
Name: Arduino USB Driver
Publisher: Arduino LLC
Always trust software from "Arduino LLC".
Install
Don't Install
You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

Passo 9 – Com a instalação terminada, podemos clicar em Fechar (Close), conforme mostrado abaixo:

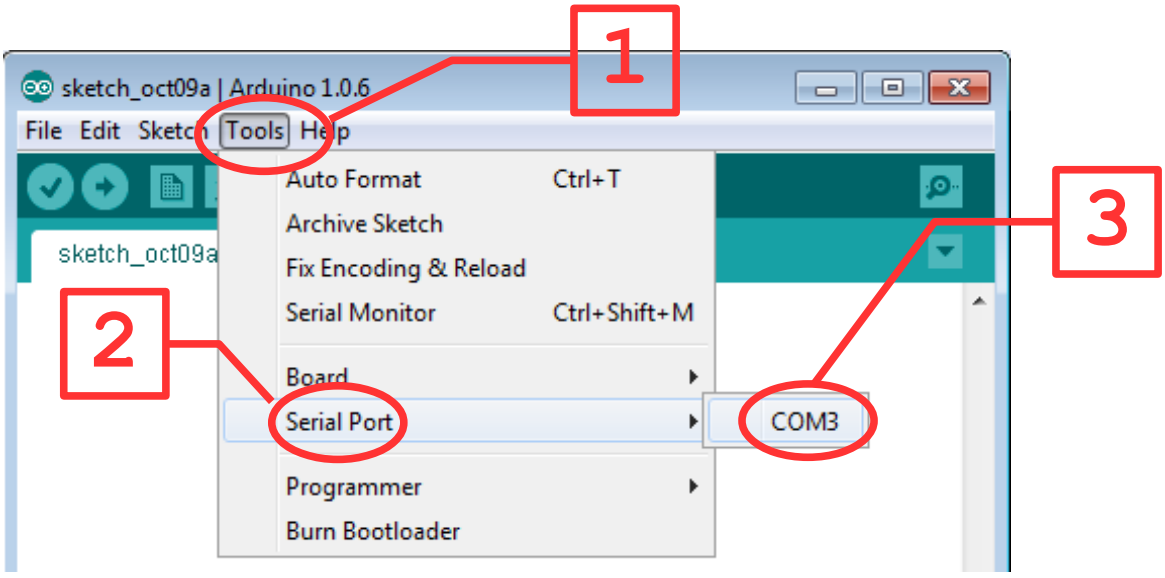


Passo 10 – Clique então no botão do Windows, e depois, no menu principal, escolha o programa recém instalado, clicando no ícone do Arduino.



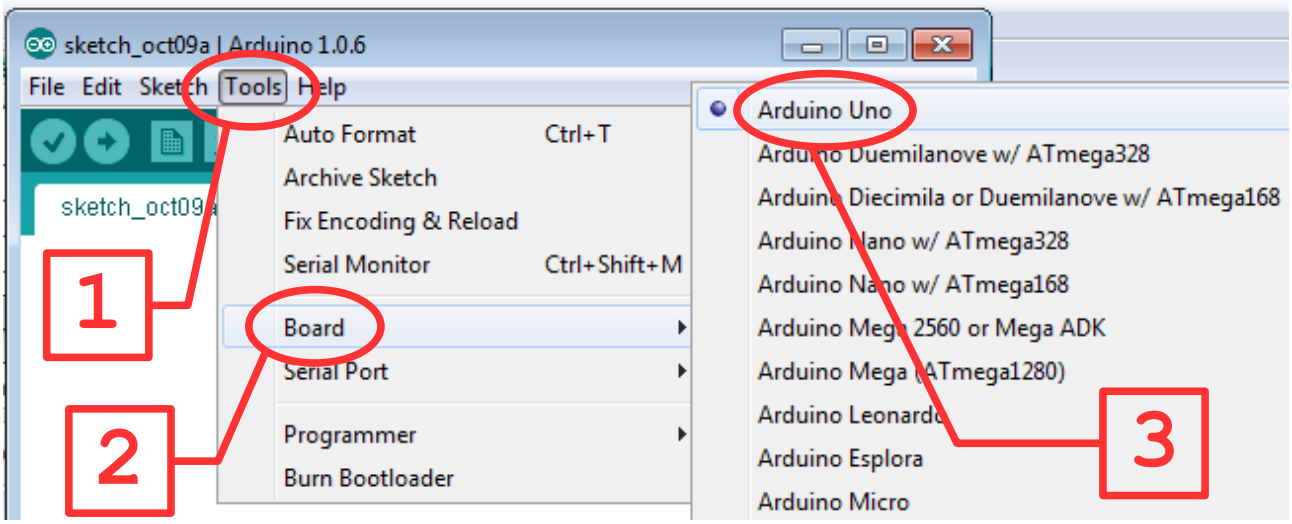
1. Clique no botão do windows
2. Localize o ícone do Arduino e clique nele

Passo 11 – Com o Arduino IDE aberto, podemos verificar se tudo deu certo, com o Arduino FTDi Basic conectado, acesse o menu “Tools”, depois passe o mouse sobre o item “Serial Port”. Se aparecer a COMX, onde X é o número da COM criada no seu computador, quer dizer que está tudo certo!



1. Clique no menu Tools
2. Passe o mouse sobre o item Serial Port
3. Clique na COMX do seu Arduino FTDi Basic

Passo 12 – Selecione a Board Arduino UNO, conforme mostrado abaixo.

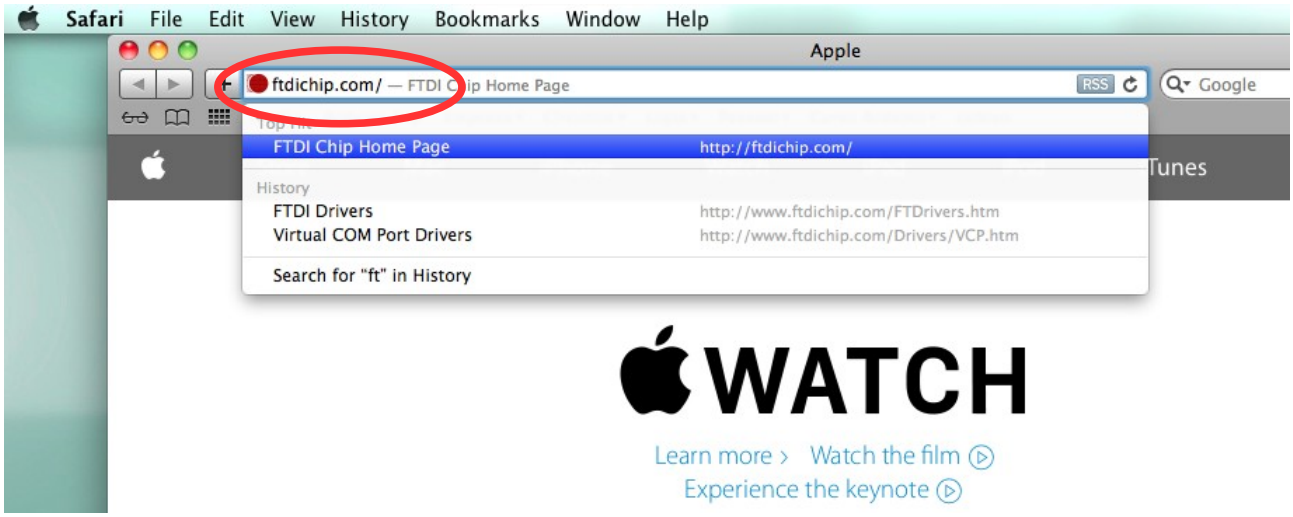


1. Clique no menu Tools
2. Passe o mouse sobre o item Board
3. Clique no item “Arduino UNO”

Instalando Drivers do FTDi Basic no MAC OS X

No MAC OS X, a instalação também é bem simples, bastando apenas seguir algumas telas de instalação e teremos tudo resolvido em pouco tempo.

Passo 1 – Acessando Site FTDi Chip

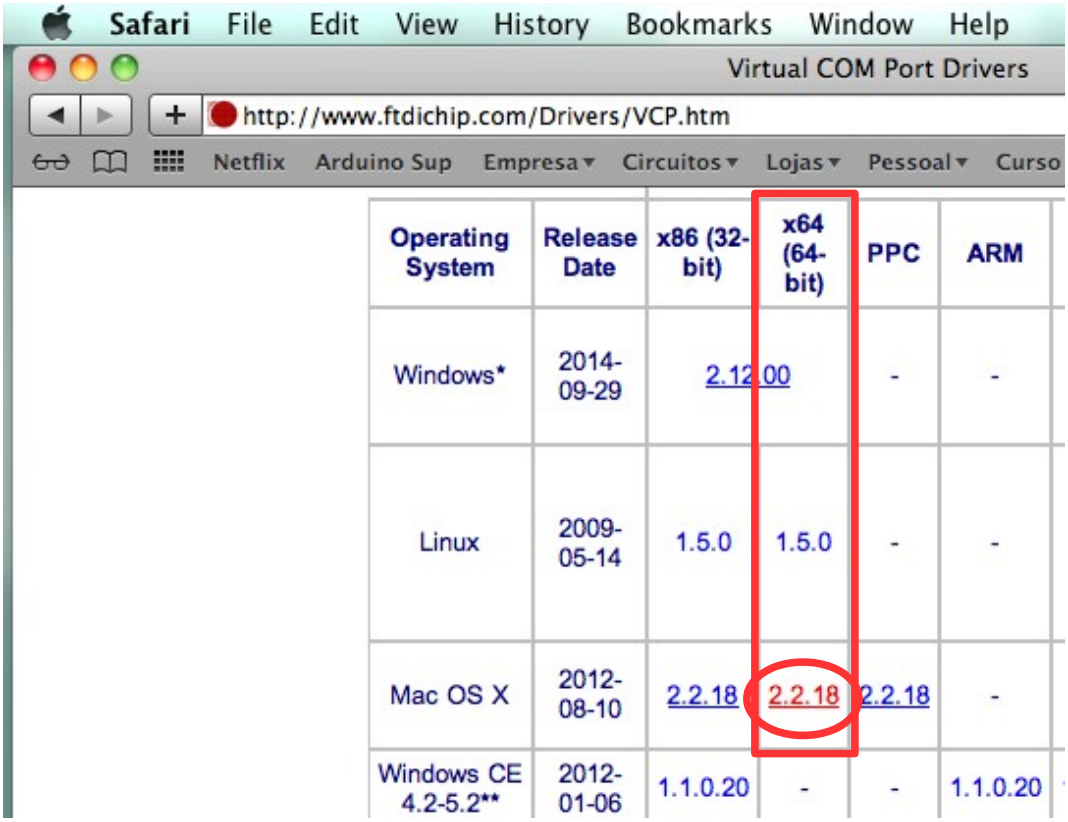


Abra seu navegador de preferência e acesse o link http://www.ftdichip.com

Passo 2 – Acessando a Página de Downloads

1. Clique na opção Drivers do menu lateral
2. Depois clique em VCP Drivers, no sub-menu que aparecer.

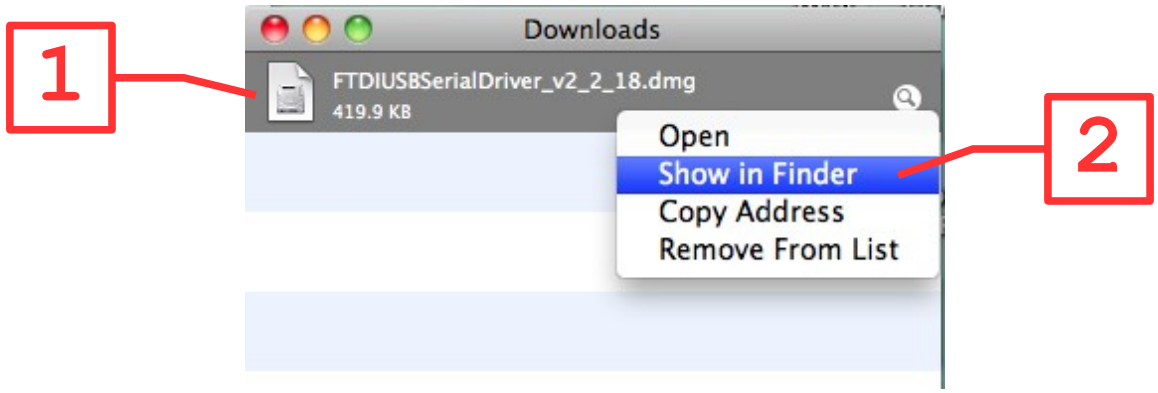
Passo 3 – Selecione a Versão do Download


Safari File Edit View History Bookmarks Window Help
Virtual COM Port Drivers
http://www.ftdichip.com/Drivers/VCP.htm
Netflix Arduino Sup Empresa Circuitos Lojas Pessoal Curso

Operating System	Release Date	x86 (32-bit)	x64 (64-bit)	PPC	ARM
Windows*	2014-09-29	2.12.00		-	-
Linux	2009-05-14	1.5.0	1.5.0	-	-
Mac OS X	2012-08-10	2.2.18	2.2.18	2.2.18	-
Windows CE 4.2-5.2**	2012-01-06	1.1.0.20	-	-	1.1.0.20

No Mac, saber se o sistema é 32 bits ou 64 bits é uma arte! Uma regra básica é saber que se o processador for Intel Core 2 Duo (99% das maquinas MAC, baseadas em Intel), ou superior, ele é 64 bits! Em resumo baixe a versão 64 bits! Se algo der errado, tente a versão 32 bits.

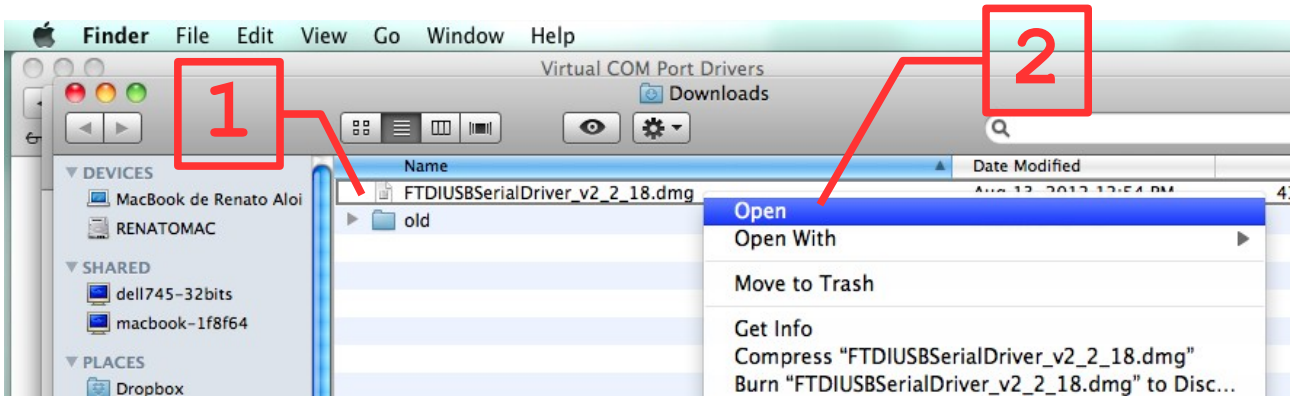
Passo 4 – Mostrando no Finder



Terminado o Download, vamos abrir o arquivo no Finder (neste momento o instalador pode apenas clicar duas vezes no arquivo baixado, da lista de Downloads).

1. CTRL + Clique para abrir o menu de contexto
2. Clique no item “Show in Finder” (Mostrar no Finder)

Passo 5 – Montando o DMG

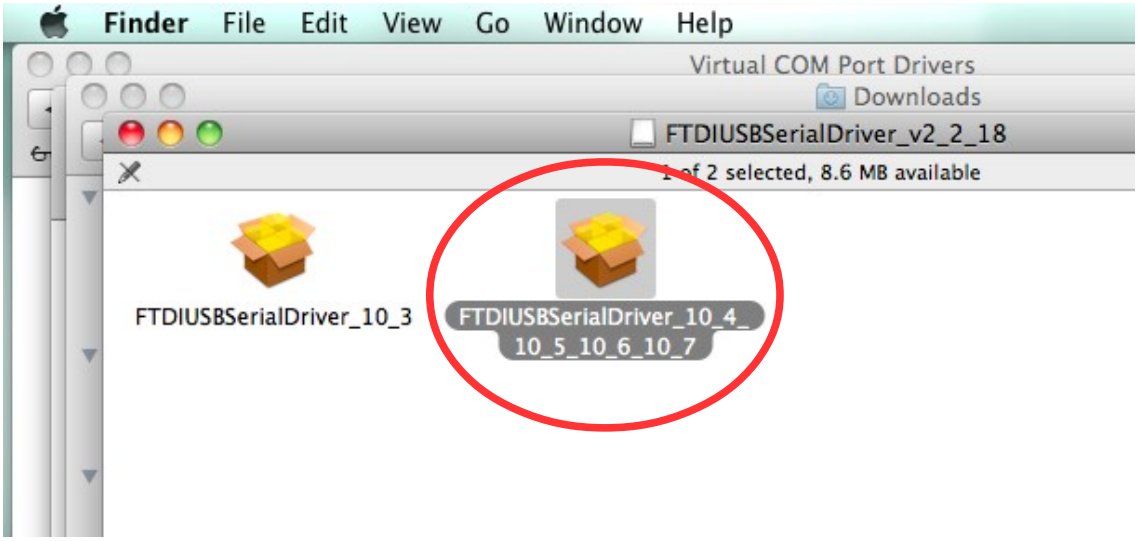


No MAC, precisamos montar o arquivo DMG em um Drive Virtual para efetuarmos a instalação. Vamos seguir as tarefas abaixo:

1. CTRL + Clique no nome do arquivo, para abrir o menu de contexto
2. Clique na opção “Open” (Abrir)

O efeito será o mesmo se o instalador apenas clicar duas vezes no nome do arquivo.

Passo 6 – Instalando Driver FTDi



Repare que no pacote temos dois arquivos de instalação:

- FTDIUSBSerialDriver_10_3 – Para MAC OS X versões 10.3
- FTDIUSBSerialDriver_10_4_10_5_10_6_10_7 – Para MAC OS X versões 10.4, 10.5, 10.6, 10.7

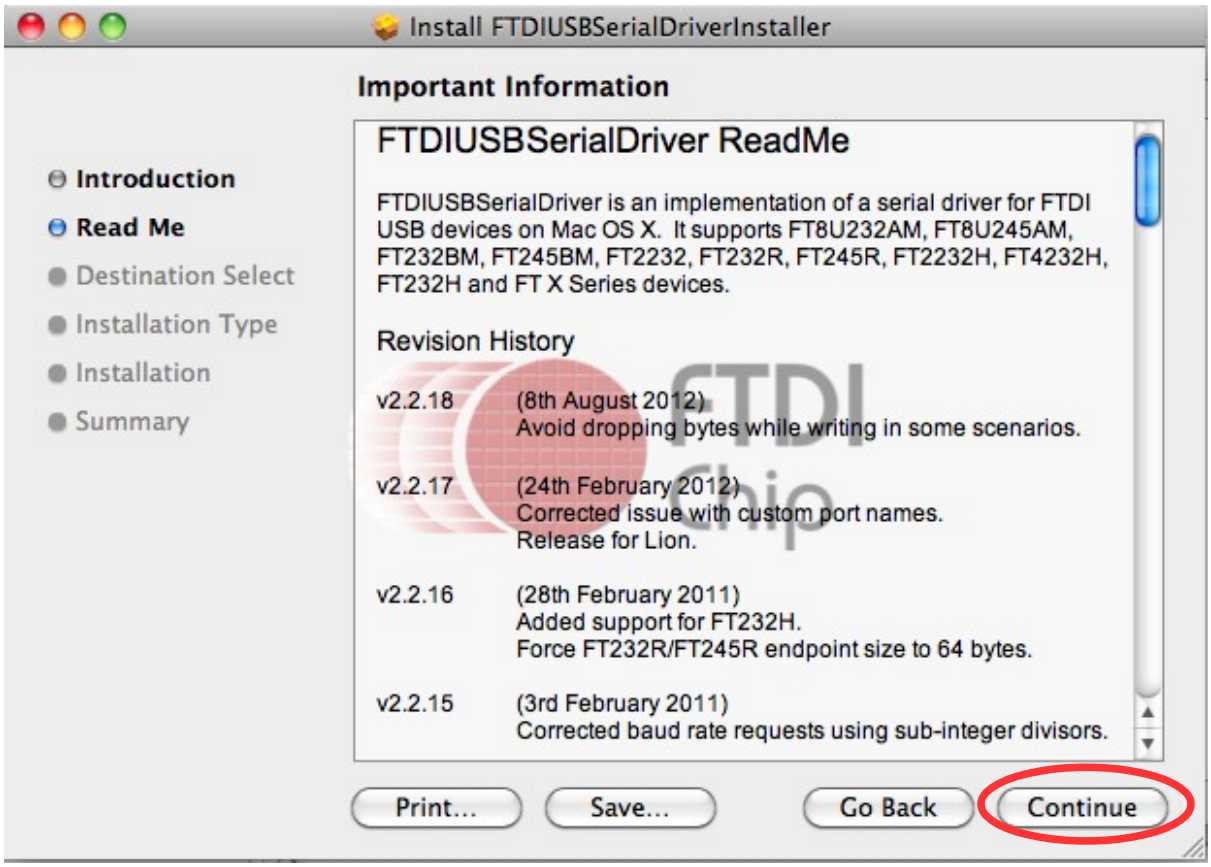
Clique duas vezes (2x) no pacote para iniciar o processo de instalação.

Atenção! Para versões anteriores ao OS X, como OS 8 ou 9, verifique a página de downloads do site FTDi Chip, pelos pacotes corretos de instalação.

Passo 7 – Introdução da Instalação do FTDi

Nesta tela, clicamos apenas em “Continue” (Continuar).

Passo 8 – Leia-me

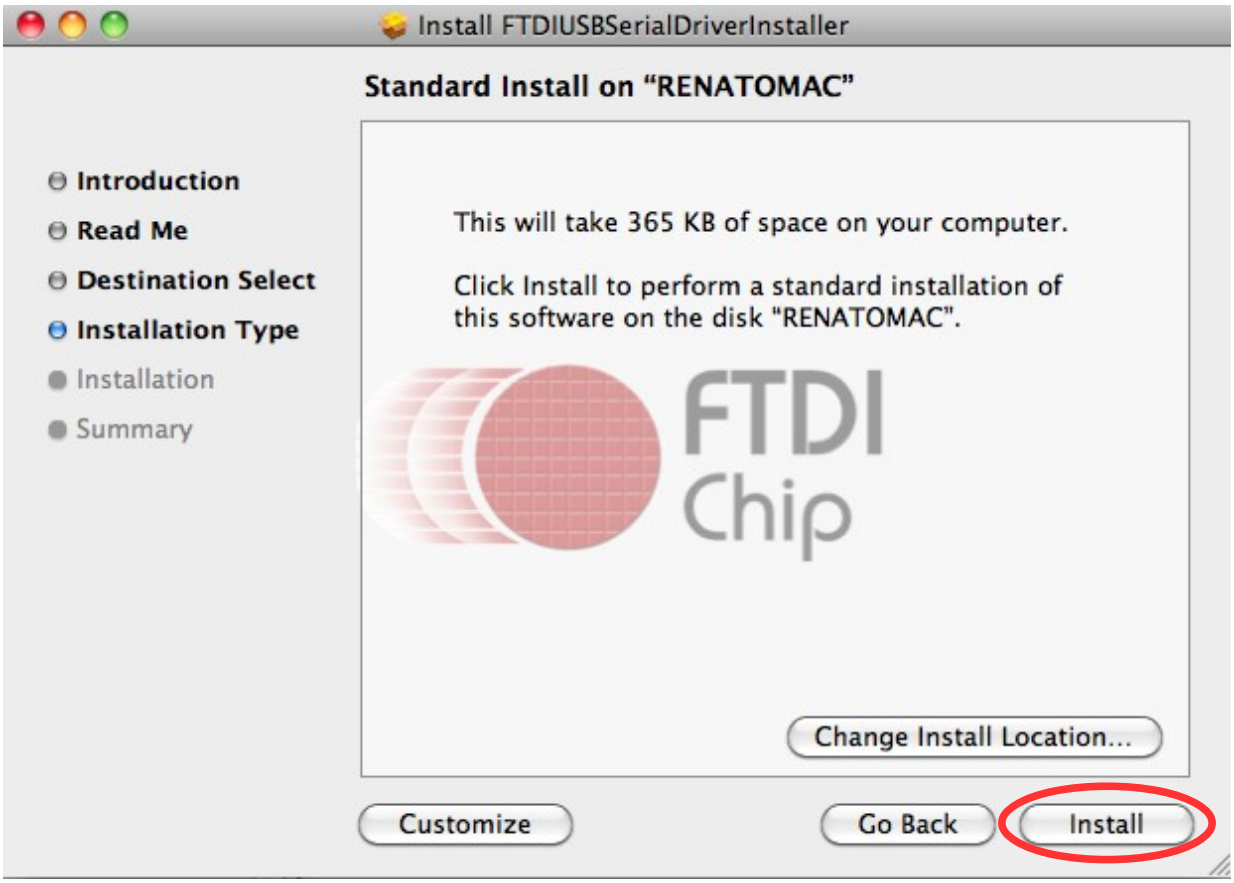

Install FTDIUSBSerialDriverInstaller
Important Information
Introduction
Read Me
Destination Select
Installation Type
Installation
Summary
FTDIUSBSerialDriver ReadMe
FTDIUSBSerialDriver is an implementation of a serial driver for FTDI USB devices on Mac OS X. It supports FT8U232AM, FT8U245AM, FT232BM, FT245BM, FT2232, FT232R, FT245R, FT2232H, FT4232H, FT232H and FT X Series devices.
Revision History
v2.2.18 (8th August 2012)
Avoid dropping bytes while writing in some scenarios.
v2.2.17 (24th February 2012)
Corrected issue with custom port names.
Release for Lion.
v2.2.16 (28th February 2011)
Added support for FT232H.
Force FT232R/FT245R endpoint size to 64 bytes.
v2.2.15 (3rd February 2011)
Corrected baud rate requests using sub-integer divisors.
Print...
Save...
Go Back
Continue

Como podemos ver, o instalador do Driver FTDi no MAC OS X é bem didático e traz muitas informações sobre as versões e notas do desenvolvimento. Mais uma vez, apenas clique em “Continue” (Continuar).

Passo 9 – Destino da Instalação

Nessa tela, normalmente também apenas clicamos em “Continue” (Continuar), pois o Disco principal já vem selecionado pela borda azul, na imagem acima.

Passo 10 – Instalando Efetivamente



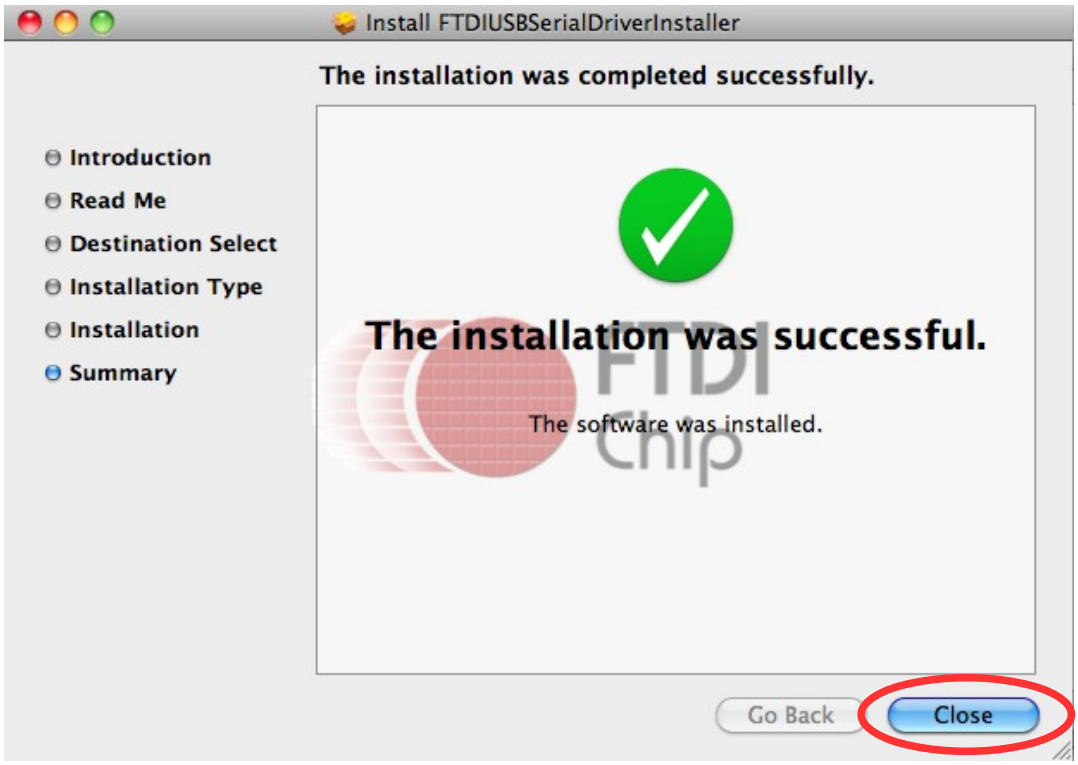
Basta clicar no botão "Install" (Instalar), conforme mostrado na imagem acima.

Passo 11 – Autorizando Instalação

Digite a senha SUDO para autorizar a instalação e clique em OK.

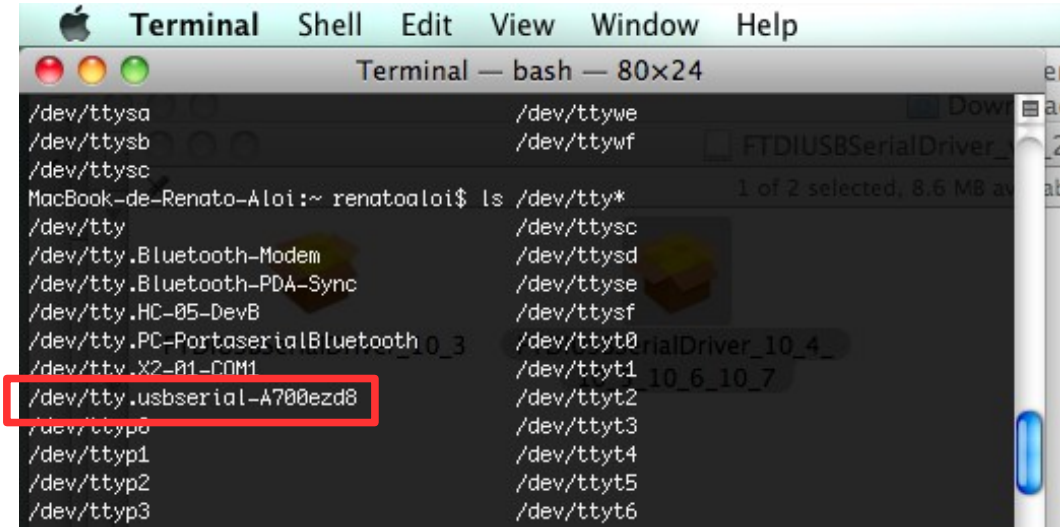
1. Digite a senha de elevação de permissões no campo Password;
2. Clique no botão OK

Passo 12 – Finalizando!



A tela final deverá se parecer com a imagem acima. Apenas clique em “Close” (Fechar), para terminar a instalação.

Passo 13 – Conferindo a instalação



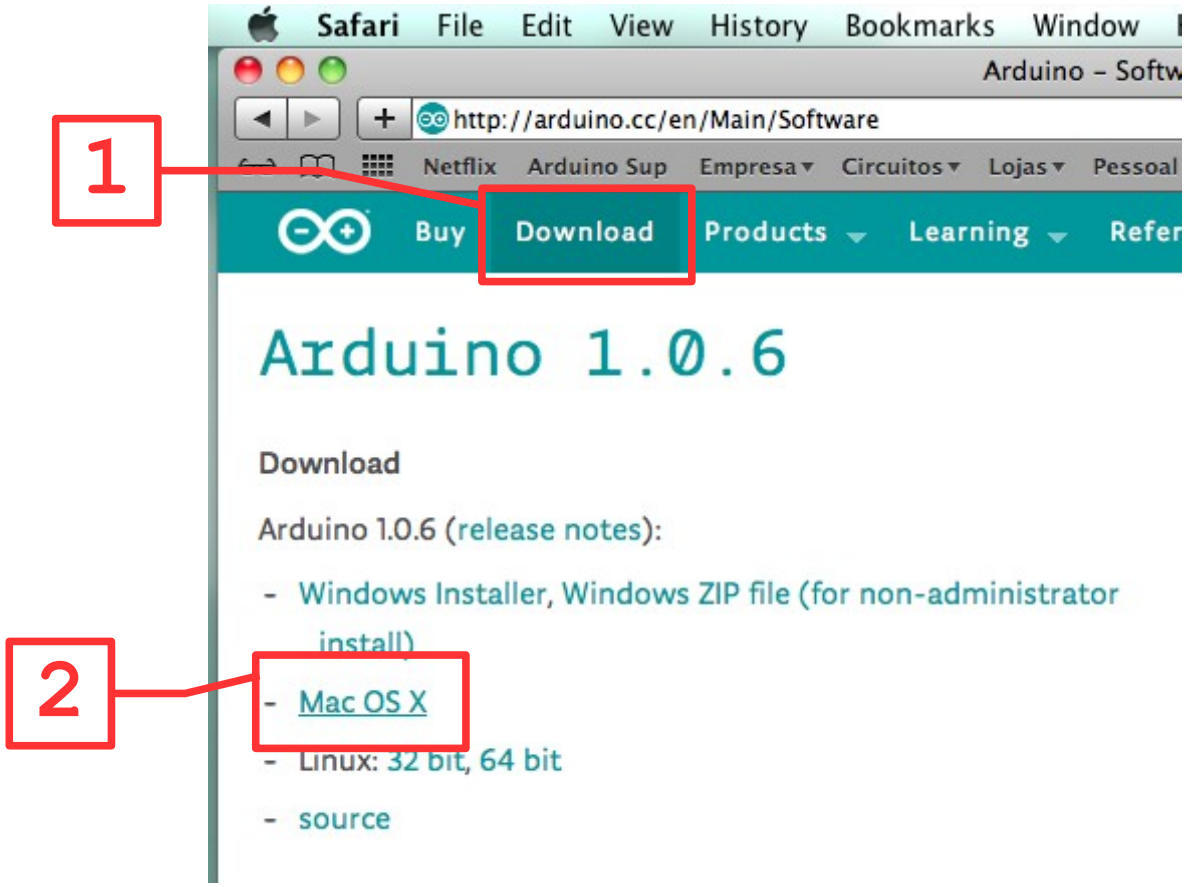
Para verificar a instalação da nova porta USB/Serial, vamos abrir um Terminal Bash e digitar o seguinte comando:

```
$ ls /dev/tty*
```

Verifique se existe um item parecido com o indicado em vermelho, na imagem acima. Esta é sua nova porta e utilizaremos a seguir na configuração do *Arduino IDE*.

Instalando o Ambiente do Arduino no MAC OS X

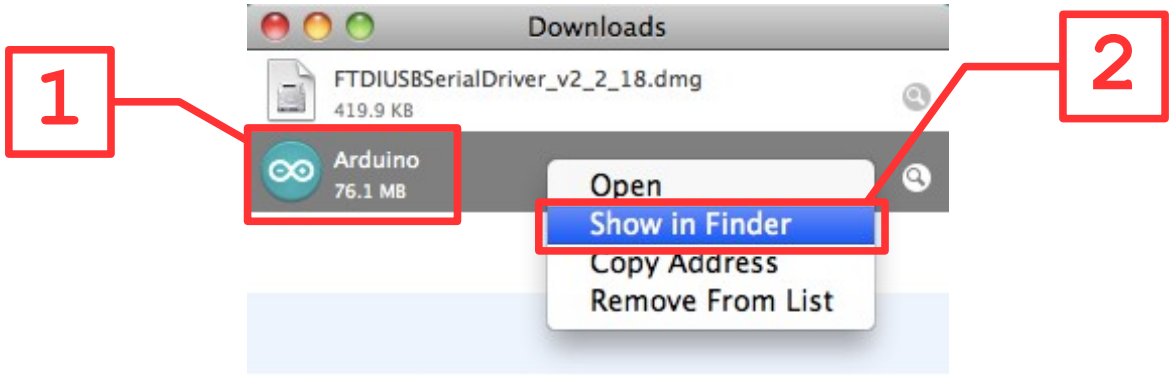
Passo 1 – Acessando Página de Download do Site do Arduino



Navegue até a página http://arduino.cc através do seu Navegador de Internet preferido.

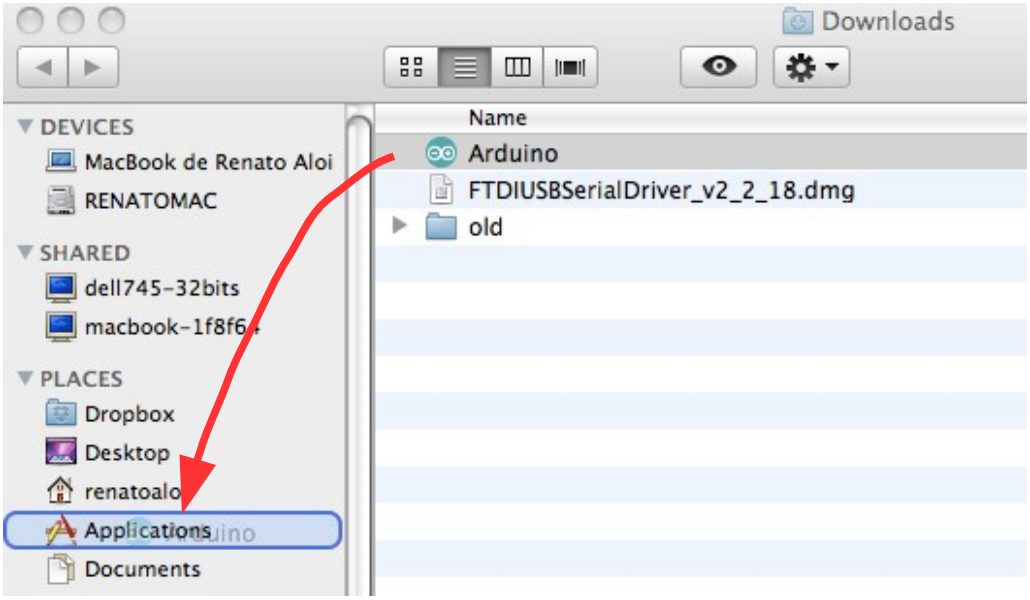
1. Clique na aba chamada Download
2. Role a página até aparecer a parte mostrada na imagem acima, depois clique no item Mac OS X

Passo 2 – Abrindo a Pasta do Download



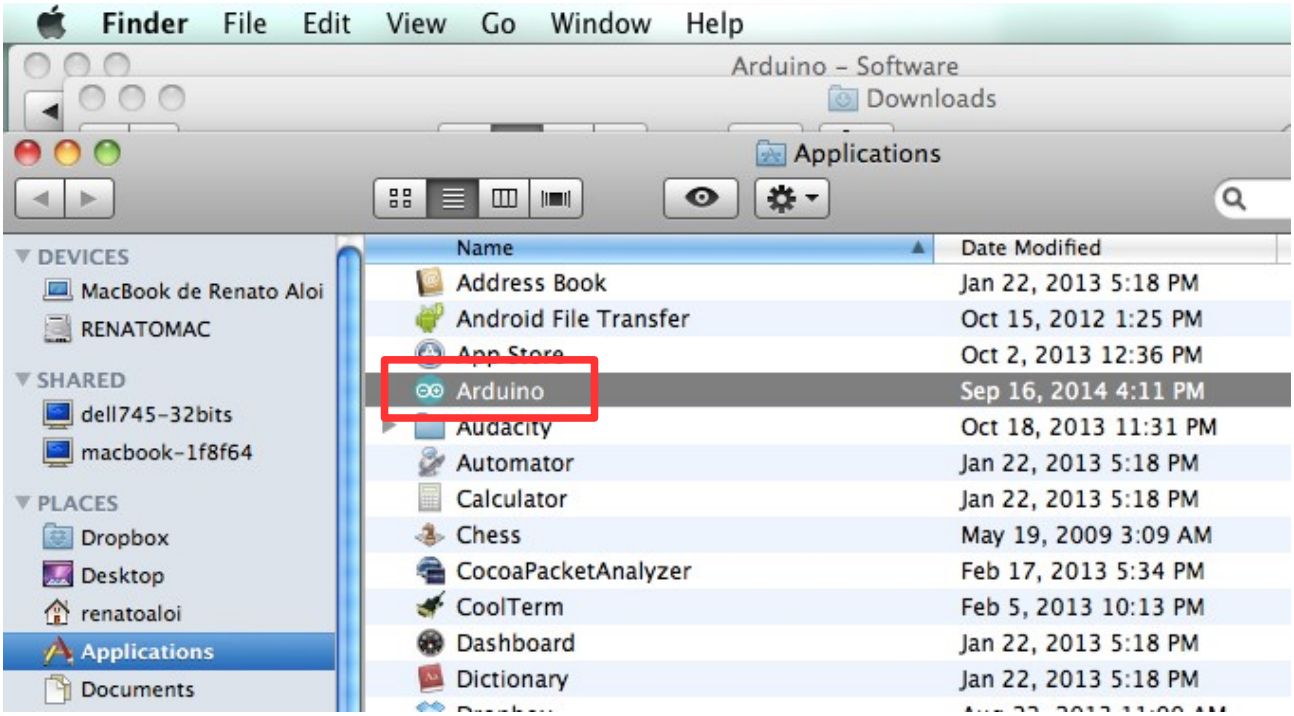
1. CTRL + Clique no item que acabamos de baixar, para aparecer o menu de contexto
2. Clique na opção “Show in Finder” (Mostrar no Finder)

Passo 3 – Arraste Arduino para Programas



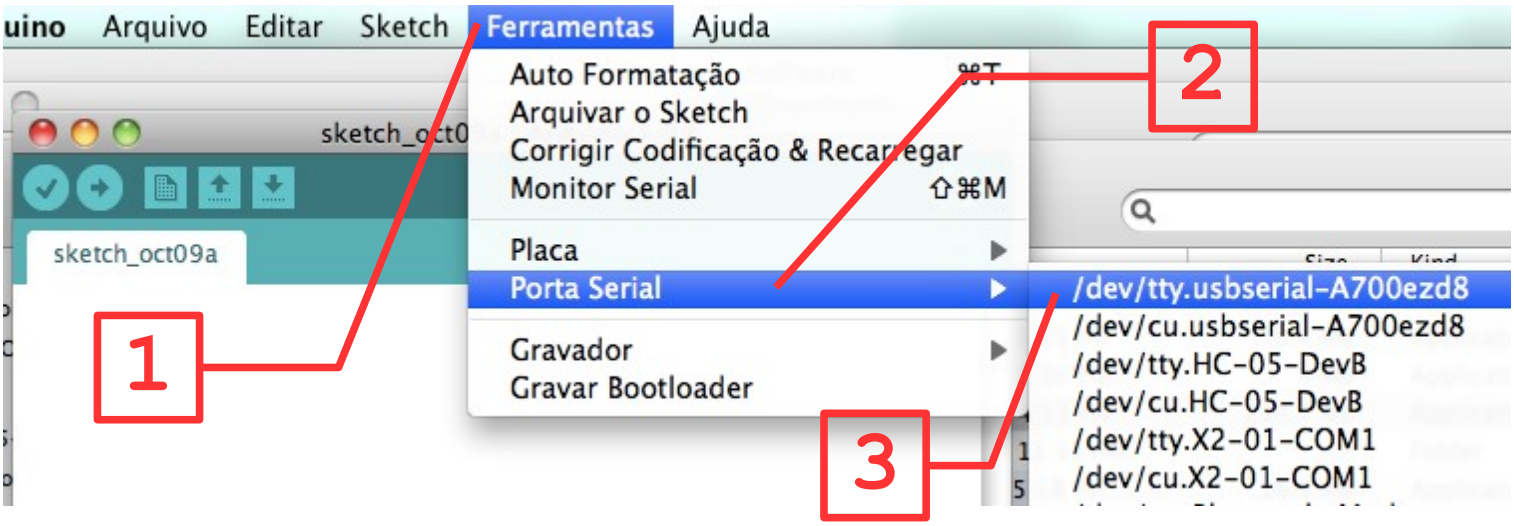
Como todo bom usuário de MAC sabe, basta arrastar o pacote Arduino para a pasta Applications (Aplicações).

Passo 4 – Arduino IDE Em Seu Devido Lugar!



Basta clicar duas vezes (2x) no ícone do Arduino para abrir o programa.

Passo 5 – Verificando a Porta Serial



1. Com o Arduino IDE aberto, clique no menu “Ferramentas” (Tools)
2. Depois passe o mouse sobre o item “Porta Serial” (Serial Port)
3. Selecione da lista a opção tty da sua nova porta serial.

Atenção! Se o computador tiver muitas portas virtuais instaladas, execute o Passo 13 da Instalação do FTDi, mais de uma vez, para comparar as listas de dispositivos com o FTDi desconectado e depois conectado. Um dos itens deve sumir e reaparecer.

Instalando Drivers do FTDi Basic no Linux

Neste manual veremos a instalação do **Driver** e do *Arduino IDE* em um *Ubuntu 14.04 32 bits*. Pela similaridade, estas instruções funcionam para a maioria das plataformas baseadas em *Debian*.

Para maiores informações ou distribuições diferentes (não *Debian*), verifique a página de instruções do próprio repositório de código da equipe Arduino, conforme segue:

https://code.google.com/p/arduino/wiki/BuildingArduino

Mas para a instalação do *Driver*, não é necessário nenhum programa adicional, basta ligar o módulo através do *Cabo USB* no *Computador* e verificar as mensagens do *Bus* da *USB*, vejamos:

Conecte o ***Arduino FTDi Basic*** no *Computador* e digite o seguinte comando em um terminal:

```
$ dmesg | tail -20
```

Se o instalador se deparar com uma tela como a mostrada abaixo, significa que tudo está *OK* e nada precisa ser feito!

```
[75161.448049] usb 4-1: new full-speed USB device number 2 using uhci_hcd
[75161.651047] usb 4-1: New USB device found, idVendor=0403, idProduct=6001
[75161.651054] usb 4-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[75161.651058] usb 4-1: Product: FT232R USB UART
[75161.651061] usb 4-1: Manufacturer: FTDI
[75161.651065] usb 4-1: SerialNumber: A700ezd8
[75161.728990] usbcore: registered new interface driver usbserial
[75161.729013] usbcore: registered new interface driver usbserial_generic
[75161.729030] usbserial: USB Serial support registered for generic
[75161.753431] usbcore: registered new interface driver ftdi_sio
[75161.753456] usbserial: USB Serial support registered for FTDI USB Serial Device
[75161.753556] ftdi_sio 4-1:1.0: FTDI USB Serial Device converter detected
[75161.753601] usb 4-1: Detected FT232RL
[75161.753604] usb 4-1: Number of endpoints 2
[75161.753607] usb 4-1: Endpoint 1 MaxPacketSize 64
[75161.753610] usb 4-1: Endpoint 2 MaxPacketSize 64
[75161.753612] usb 4-1: Setting MaxPacketSize 64
[75161.756979] usb 4-1: FTDI USB Serial Device converter now attached to ttyUSB0
renatoaloi@Del
```

Só a mensagem acima já nos garante que está tudo certo, mas vamos listar os *Device's*, através do comando:

```
$ ls /dev/tty*
```

Se no resultado aparecer a ttyUSBX (onde X é o número da porta USB), assim como listado no **dmesg**, estamos resolvidos!

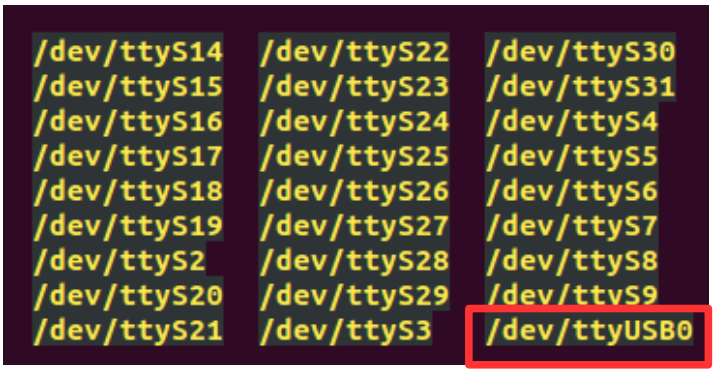


Imagem da lista de *Devices* e o ttyUSB0 aparecendo como uma porta USB/Serial disponível e pronta para uso!

Atenção! Para que a *Porta Serial* funcione no *Arduino IDE*, rodando sob o usuário normal, precisamos modificar algumas permissões, efetue as etapas a seguir para configurar as permissões do dispositivo:

```
$ sudo usermod -a -G dialout nomeusuario
```

```
$ sudo chmod a+rw /dev/ttyUSBX
```

Onde **nomeusuario** é o nome do seu usuário; e **X** é o número da porta USB, normalmente zero (0). Exemplo: */dev/ttyUSB0*

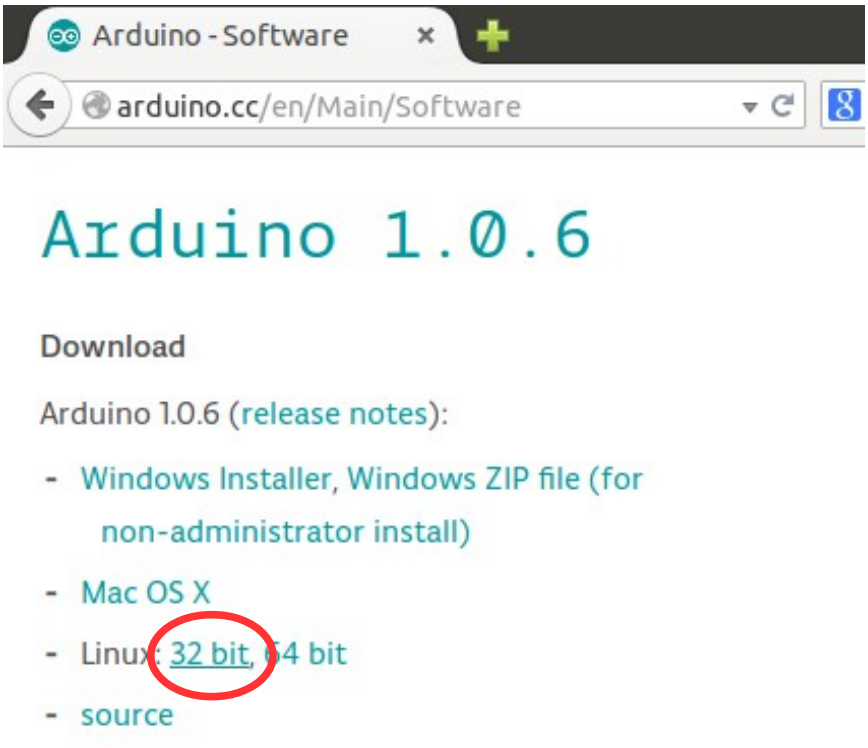
Instalando o Ambiente do Arduino no Linux

Bom, agora podemos instalar o *Arduino IDE*. Vamos fazer isso, seguindo os passos abaixo.

Passo 1 – Acessando o Site do Arduino

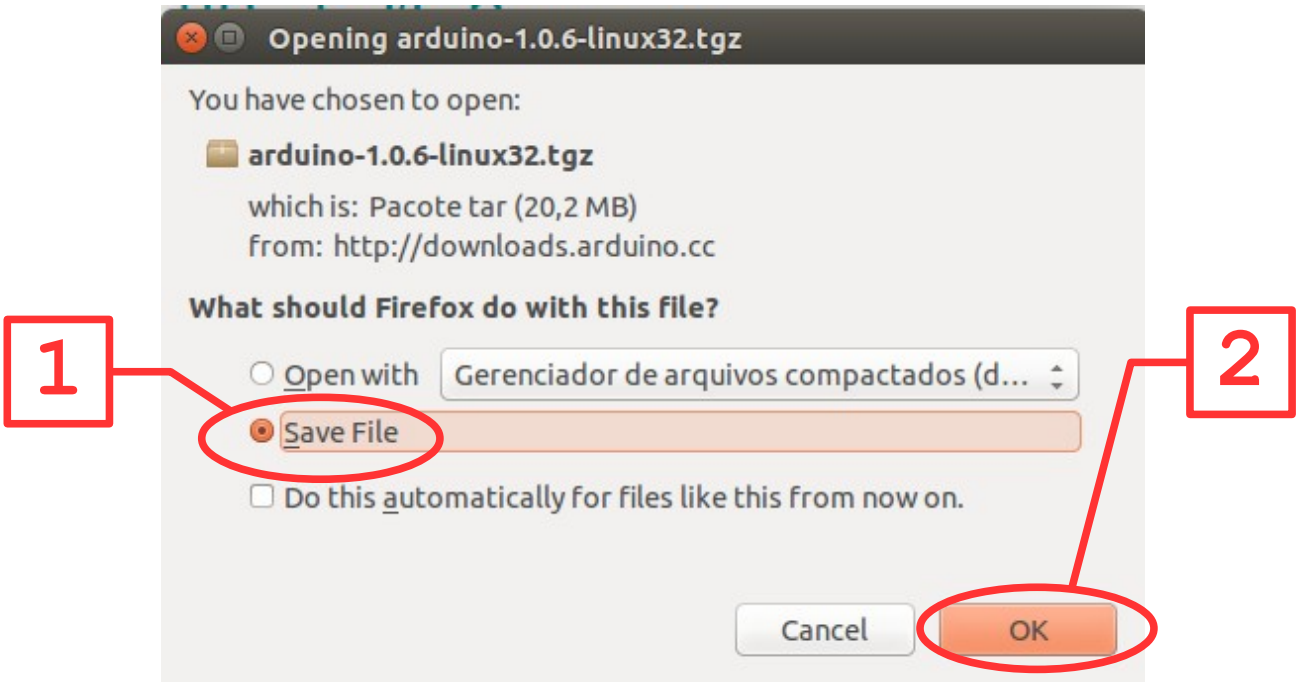
1. Clique no ícone para abrir o Navegador Firefox
2. Navegue até o endereço http://arduino.cc
3. Clique no botão “Download”, conforme mostrado acima.

Passo 2 – Selecionando a Versão de Download



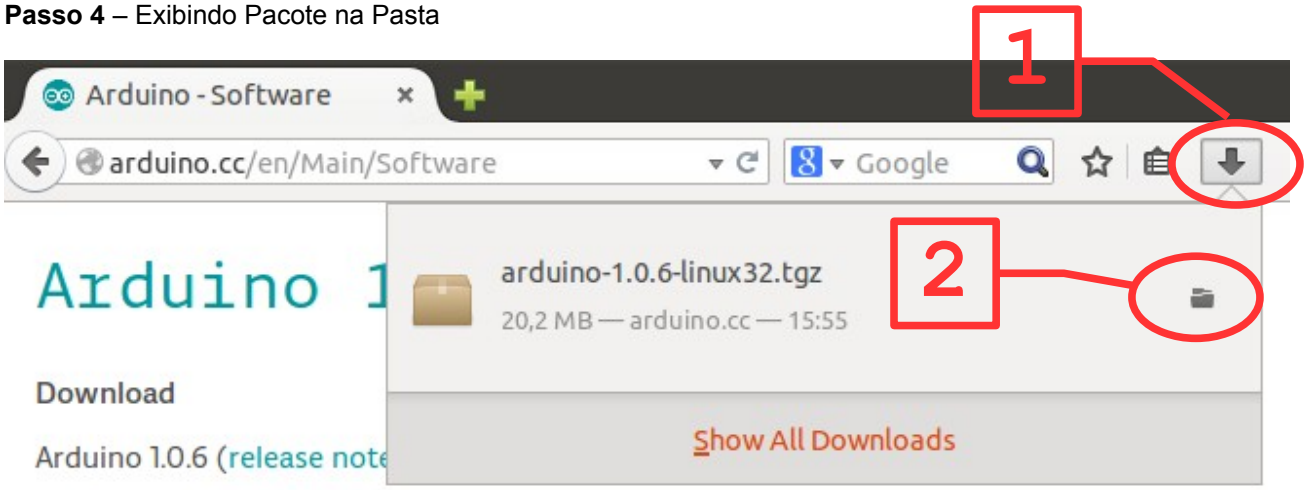
Aqui, para os usuários do Linux temos algumas opções: podemos baixar os códigos-fonte e compilá-los. Ou podemos baixar os pacotes compilados em 32 ou 64 bits. No caso deste manual, vamos baixar a versão 32 bits, clicando na opção marcada, na imagem acima.

Passo 3 – Salvando Pacote do Arduino IDE



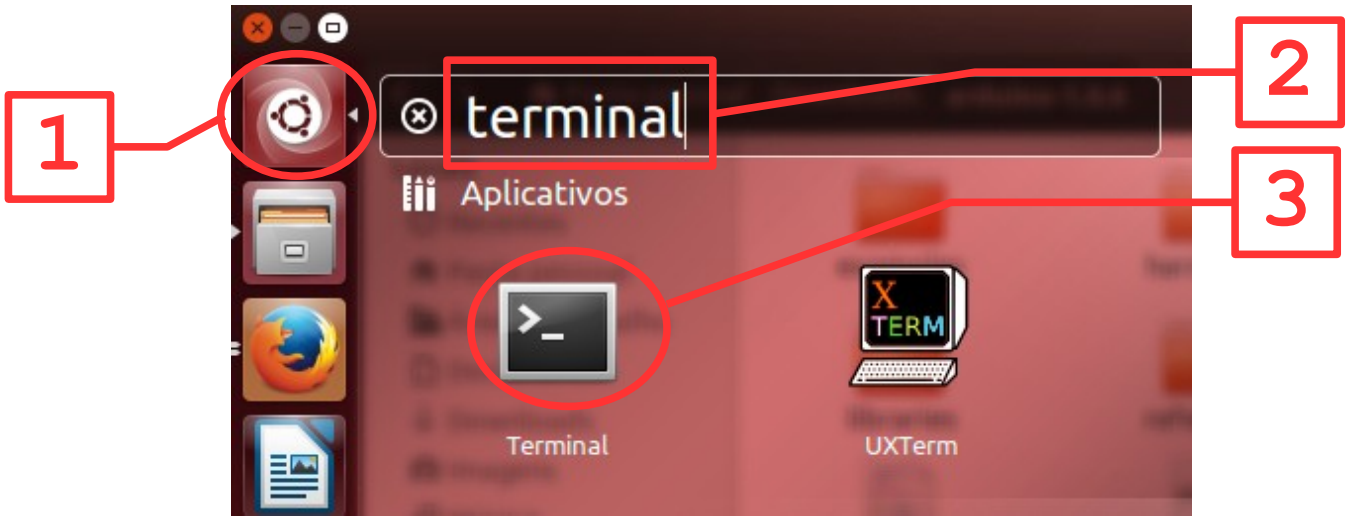
1. Selecione a opção “Save File” (Salvar Arquivo no disco)
2. Clique no botão OK

Passo 4 – Exibindo Pacote na Pasta



1. Clique no botão do gerenciador de downloads do navegador
2. Depois clique no ícone da pasta de arquivo para verificarmos o arquivo baixado

Passo 5 – Efetuando as Tarefas Pelo Terminal



1. Clique no ícone do Ubuntu, no lançador;
2. Digite “terminal” no campo de pesquisa; e
3. Clique no aplicativo Terminal que aparecer.

Bom, não seria um tutorial de Linux se não abríssemos um Terminal! Vamos continuar a instalação com comandos do Bash!

Passo 6 – Descompactando Pacote do Arduino IDE

Bom, agora que estamos no Terminal Bash, vamos começar criando um caminho melhor para nosso *Arduino IDE* ficar.

O instalador pode decidir onde instalar o pacote do *Arduino IDE*, inclusive trabalhar com versões do *Arduino IDE* diferentes, no mesmo computador.

Mas para a instalação padrão, o mais comum é criar uma pasta sob **`/usr/share`** e fazer os links simbólicos necessários para a pasta de executáveis: **`/usr/bin`**.

É isso que vamos fazer nesse manual, primeiro criamos a pasta com o comando:

```
$ sudo mkdir /usr/share/arduino
```

Depois, o comando descompactar, com opção para enviar os arquivos para a pasta que acabamos de criar:

```
$ cd ~/Downloads
$ sudo tar -xvzf arduino-1.0.6-linux32.tgz -C /usr/share/arduino
```

Passo 7 – Instalando Pacotes do Java

Um requisito para o funcionamento do *Arduino IDE* é o ambiente do *Java*, instalado. Precisamos do *Java Runtime Enviroment (JRE)*. Vamos instalar através do **`aptitude`**:

```
$ sudo apt-get install openjdk-7-jre
```

Passo 8 – Criando Link Para Executável

O arquivo executável do *Arduino IDE* na verdade é um script Bash, para executar algumas tarefas, inclusive lançar o *Java* com as configurações do *Arduino IDE*.

Então só o que precisamos fazer é criar um link simbólico para o *script*, em algum lugar que esteja no *PATH* e fácil de acessar. Escolhemos o destino: **`/usr/bin`**, então vamos ao comando:

```
$ sudo ln -s /usr/share/arduino/arduino-1.0.6/arduino /usr/bin/arduino
```

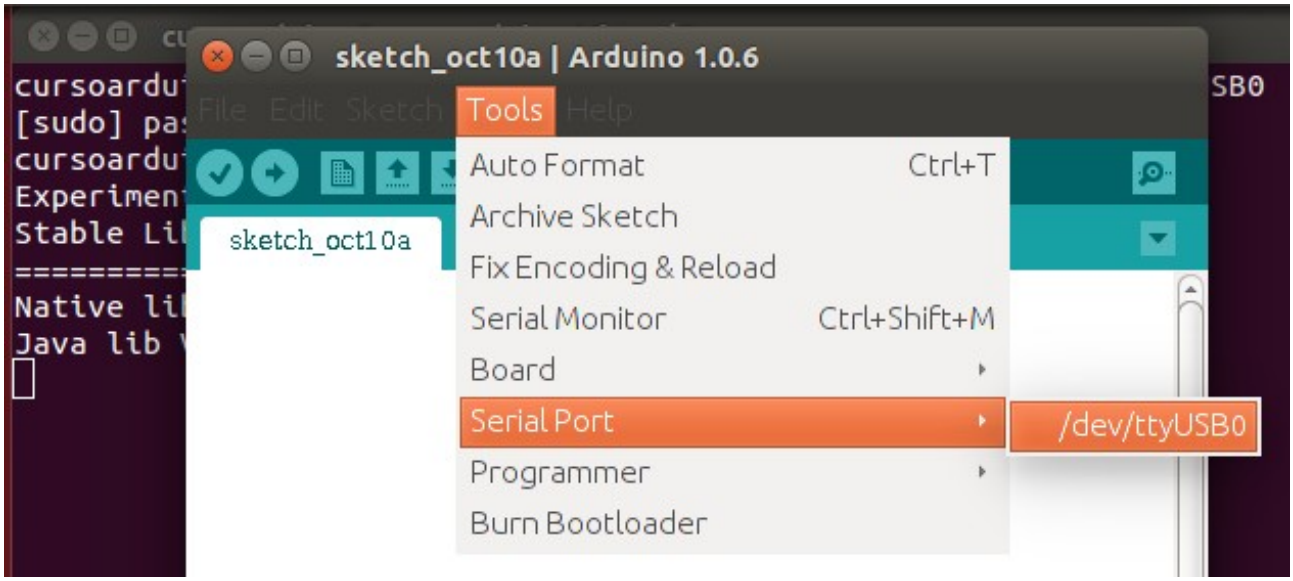
Passo 9 – Executando o Arduino IDE

Se todas as configurações de segurança deram certo, ao digitar **`arduino`** na linha de comando do Bash, teremos uma tela parecida com a mostrada abaixo.

Atenção! Repare que a opção *"Serial Port"* está habilitada! Se o instalador, lá, nas últimas etapas da Instalação do Driver FTDi, não fizer as configurações certas, aqui, nesse momento, o menu *"Serial Port"* estará acinzentado e indisponível. Verifique as permissões do usuário atual. Se isso acontecer significa que o usuário atual está sem permissões de acesso a porta serial **`/dev/ttyUSB0`**, ou que ela não está no modo de leitura e escrita. Revise os últimos passos da instalação do Driver do FTDi!

Para iniciar o *Arduino IDE*, execute o seguinte comando no Bash:

```
$ arduino
```



Configurando e Carregando o Painel de Controle Web

No tópico anterior, cobrimos todas as instalações possíveis dos *Drivers* e programas essenciais. Daqui para frente vamos aprender a instalar e configurar o ***Painel de Controle Web***, programa exclusivo do ***Kit Curso Arduino Automation***, que roda através da biblioteca *EthernetAutomation*. Veremos mais detalhes a seguir.

Já vimos no início do manual uma imagem da aparência do ***Painel de Controle Web***, com seus botões grandes e visual agradável. Assim não precisamos ficar redimensionando a página com os dedos, toda vez que ela recarrega!

Várias funcionalidades adicionais vem sendo implementadas na biblioteca *EthernetAutomation*, desde seu lançamento, e não existe previsão de acabar. Sempre teremos novidades e melhorias! Basta acompanhar no repositório de códigos-fonte, pelo link:

https://github.com/renatoaloi

Para o ***Kit Curso Arduino Automation***, precisamos acompanhar dois projetos de biblioteca:

- **EthernetAutomation** – Biblioteca responsável por apresentar a página do *Painel de Controle Web*
- **EtherEncLib** – Biblioteca especialmente criada para o Módulo Ethernet ENC28J60. Uma poderosa ferramenta para conectar nosso equipamento na Ethernet/Internet!

Atenção – A instalação das bibliotecas mostradas a seguir considera que você já está com o seu ***Arduino FTDI Basic*** devidamente conectado no ***Arduino Supervisório REV2***. Mas qual? Essa é a parte importante! Essas bibliotecas devem ser carregadas no *Arduino Supervisório* ***#1***, que está conectado no *Módulo Ethernet Enc28J60*.

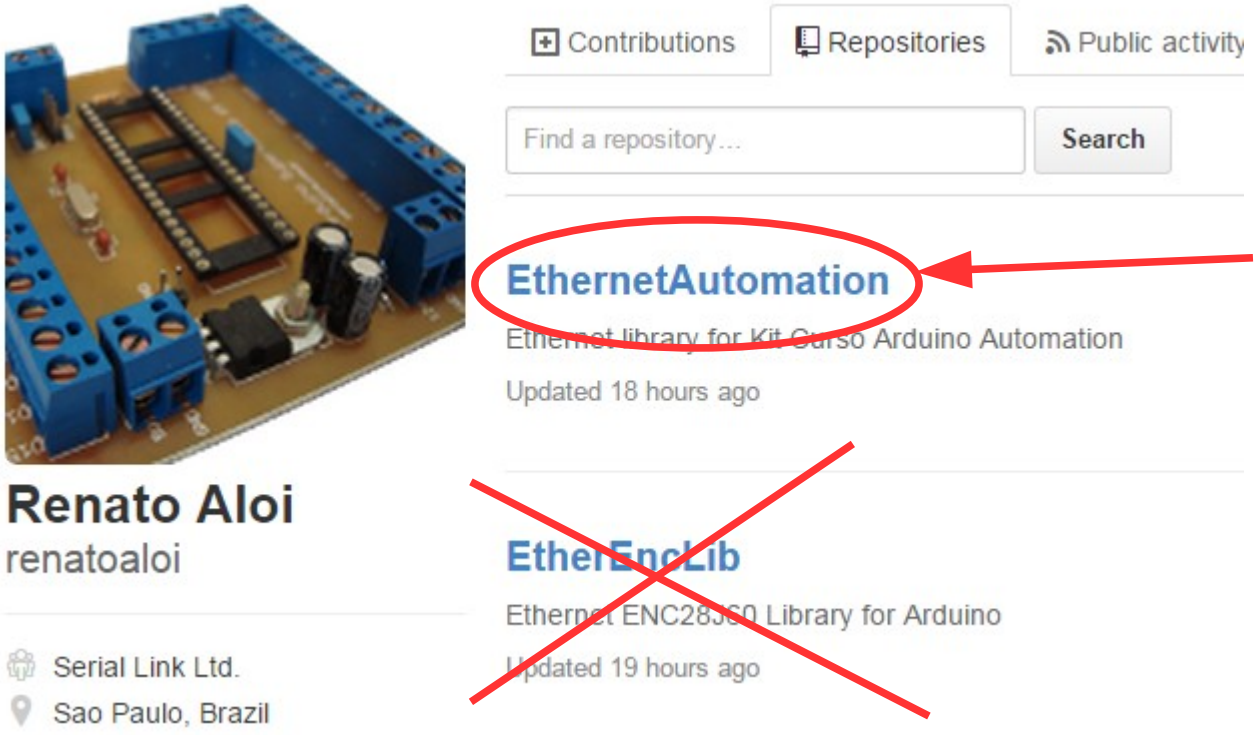
Instalando Bibliotecas Automation no Windows

Vamos ver as instalações agora das bibliotecas ***EthernetAutomation*** e ***EtherEncLib*** no ambiente Windows.

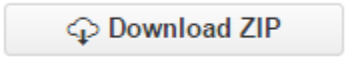
Passo 1 – Abrir site do GitHub, acessando repositório de Renato Aloi.

1. Acessar site https://github.com/renatoaloi
2. Clicar na aba “Repositories” (Repositórios)

Passo 2 – Clique no repositório *EthernetAutomation*



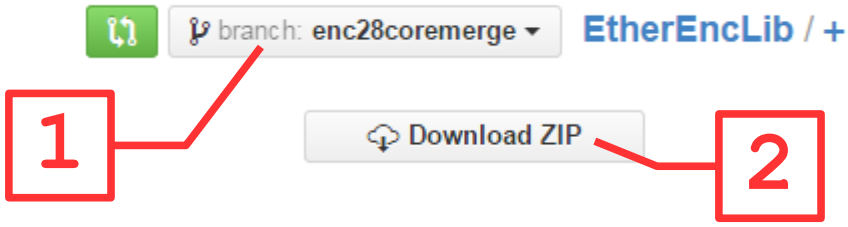
Passo 3 – Na página que aparecer, clique no ícone mostrado abaixo, para fazer o *Download* do *Arquivo ZIP*:



Passo 4 – Depois de fazer o *Download* do *Passo 3*, desça a página e clique no link indicado na figura abaixo:

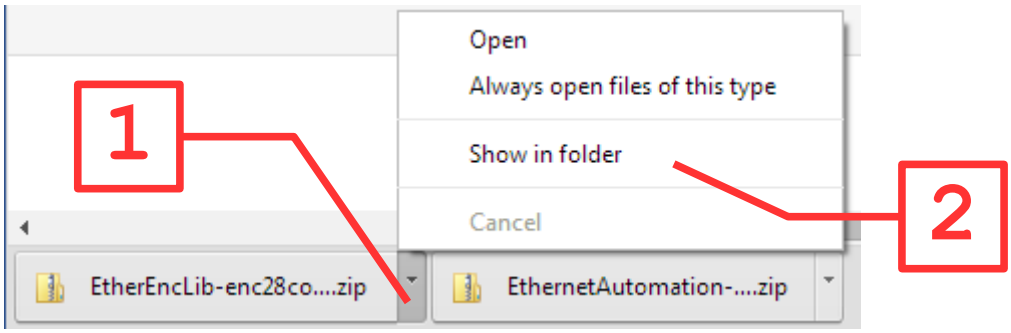
1. Clique no link: https://github.com/renatoaloi/EtherEncLib/tree/enc28coremerge

Passo 5 – Clique também no botão de *Download* do *Arquivo ZIP* dessa biblioteca:



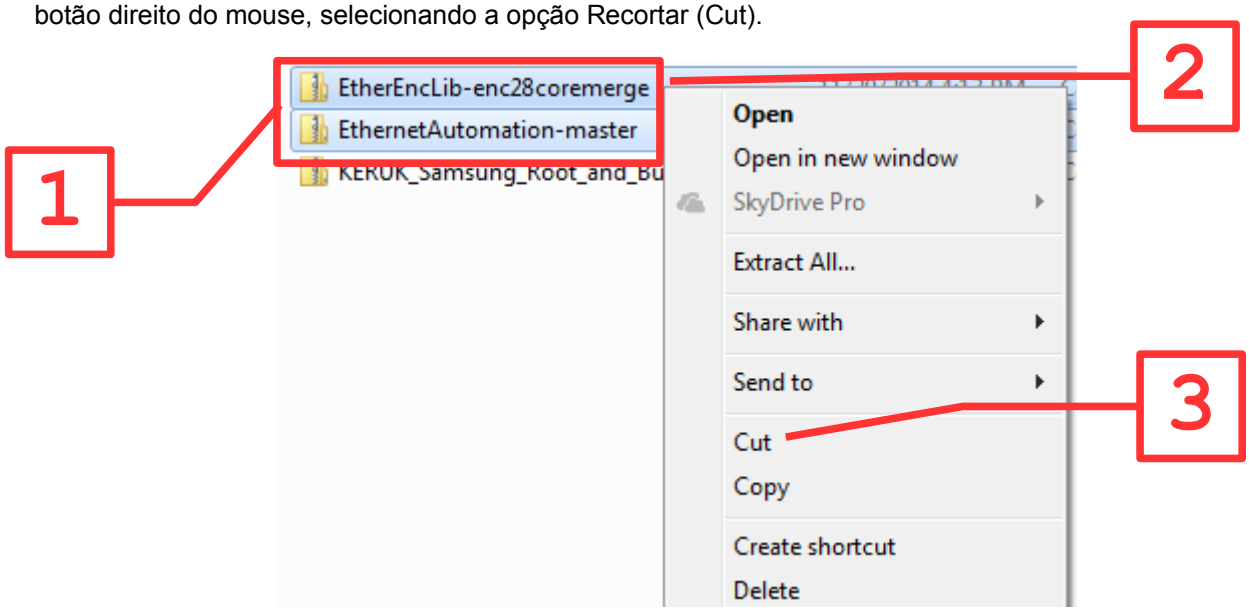
1. Verifique que o ramo (branch) selecionado é o **`enc28coremerge`**
2. Clique no botão *Download ZIP*

Passo 6 – Escolha um dos arquivos que foi feito *Download* e clique na opção "*Show In Folder*" (Mostrar na Pasta), conforme mostrado abaixo:



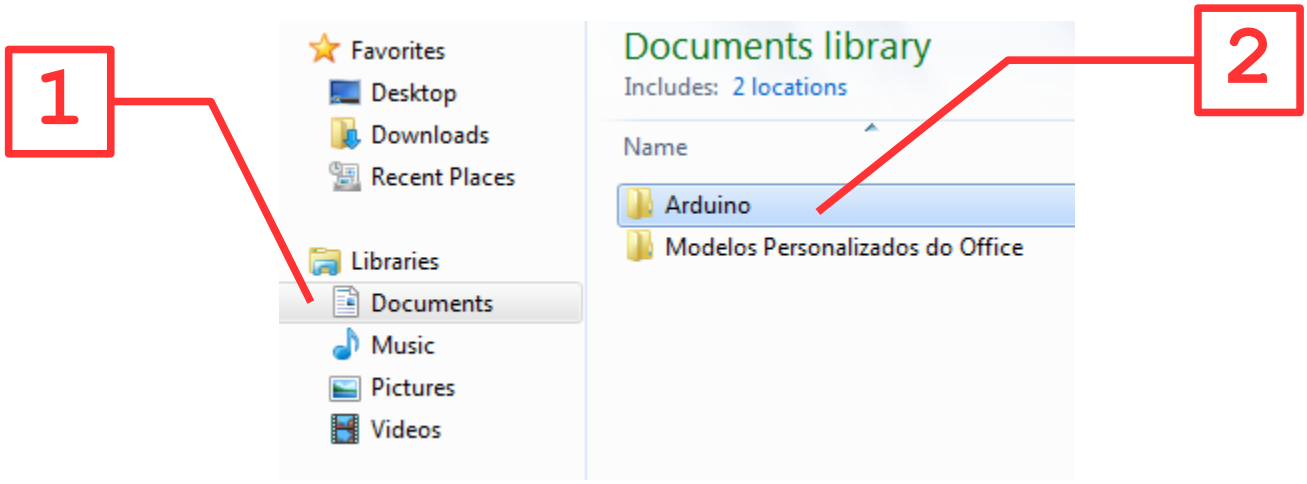
1. Clique na área da seta, indicada na imagem
2. No menu que aparecer, clique em "*Show in folder*" (Mostrar na pasta)

Passo 7 – Na pasta de *Downloads*, selecione os dois arquivos que acabamos de baixar e clique com o botão direito do mouse, selecionando a opção Recortar (Cut).



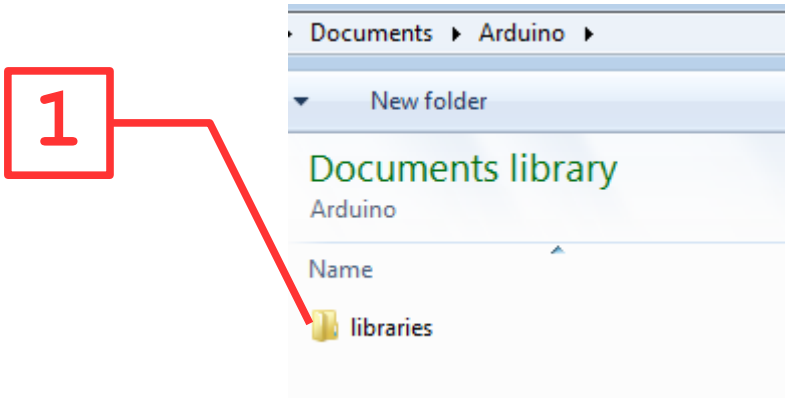
1. Selecione os dois arquivos, arrastando o ponteiro do mouse
2. Clique com o botão direito do mouse em qualquer um dos arquivos
3. Selecione "*Cut*" (Recortar) do menu que aparecer

Passo 8 – Vamos navegar até a pasta de *Sketches* do *Arduino*, siga as etapas abaixo:



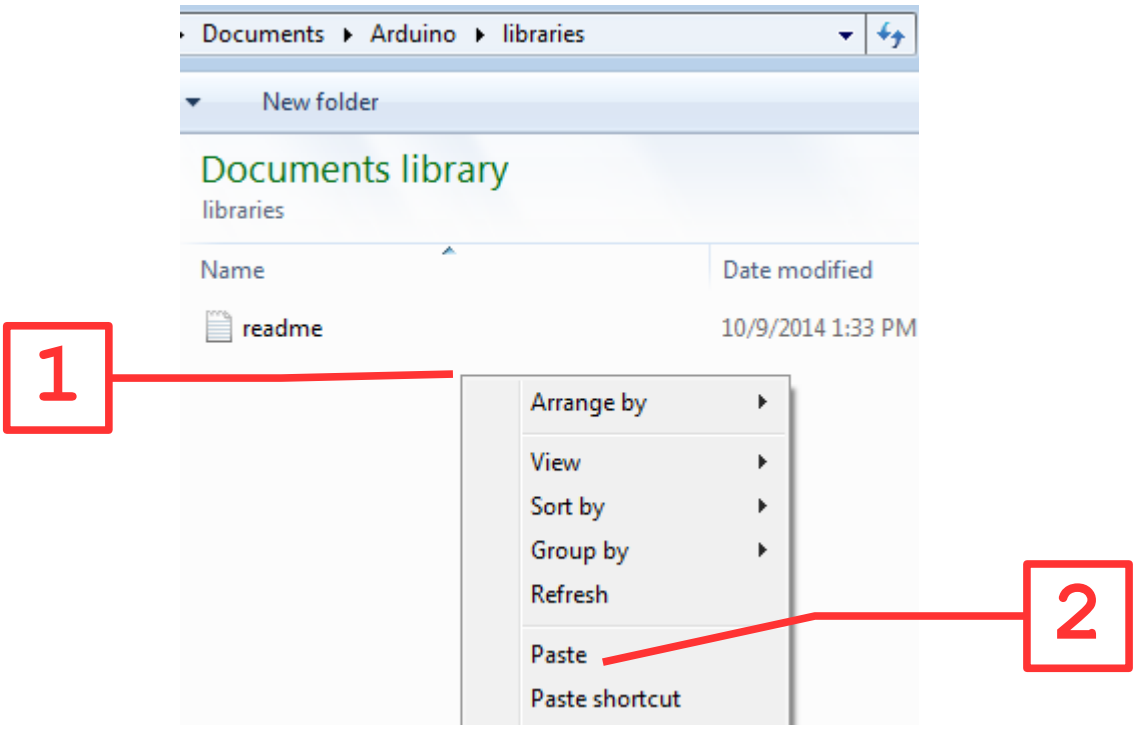
1. Clique na pasta "*Documents*" (Meus Documentos)
2. Clique 2x na pasta *Arduino* (local onde são salvos nossos códigos-fonte)

Passo 9 – Finalmente, clique na pasta *libraries*, que é o destino final dos nossos arquivos, conforme segue:



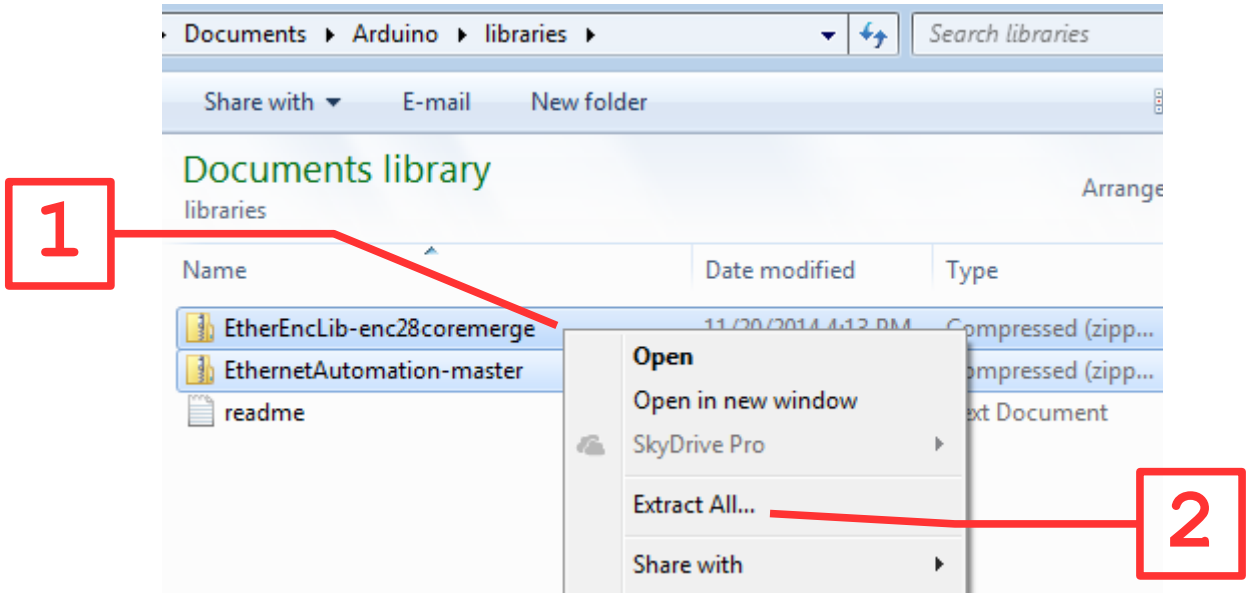
1. Clique 2x na pasta "*libraries*" (bibliotecas), dentro da pasta *Arduino*

Passo 10 – Nesta pasta que acabamos de abrir, escolha um espaço e clique com o botão direito do mouse para colar os arquivos que recortamos da pasta de *Downloads*.



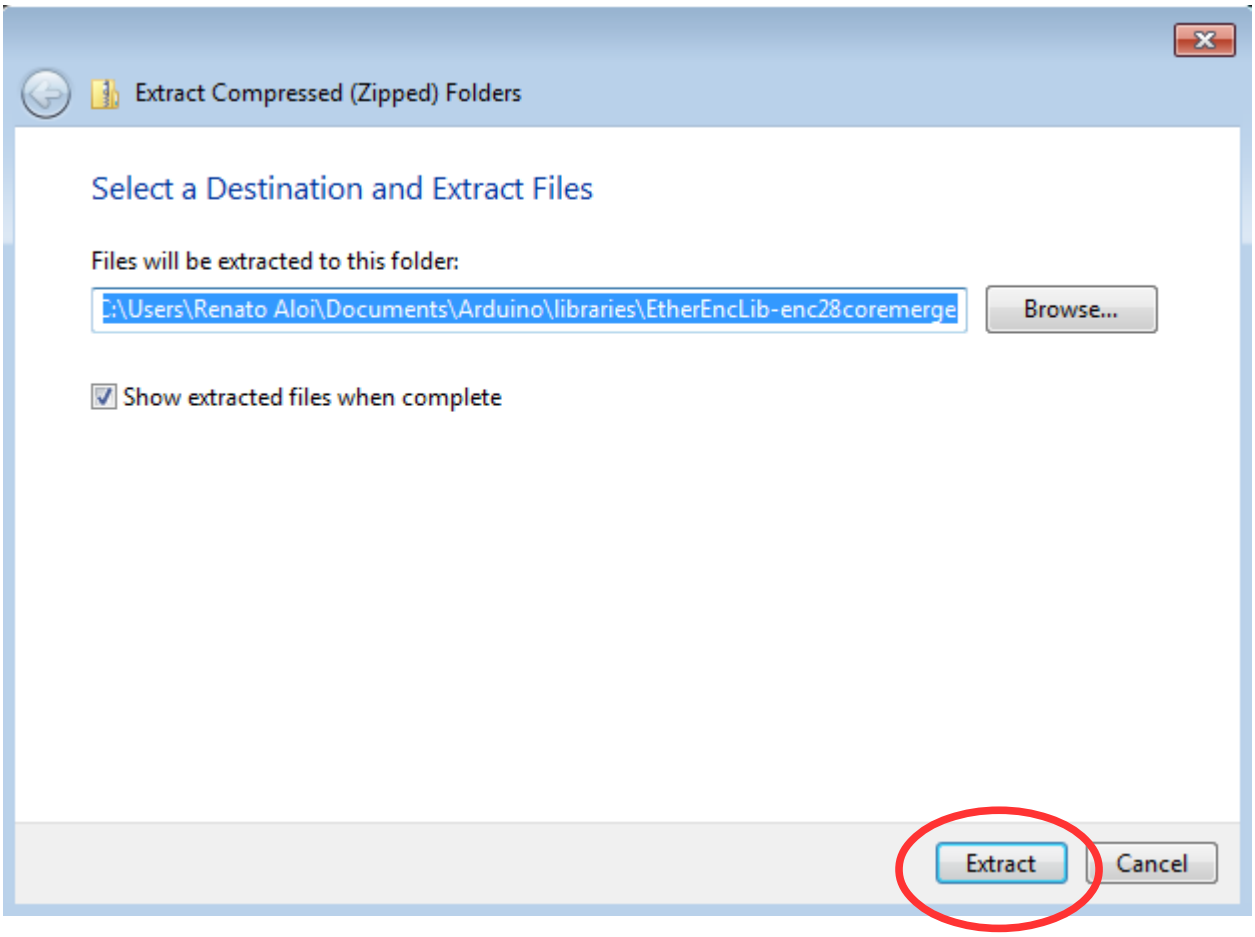
1. Selecione um local vazio dentro da pasta "*libraries*" e clique com o botão direito do mouse
2. No menu de contexto que aparecer, clique na opção "*Paste*" (Colar)

Passo 11 – Assim que os arquivos aparecerem na pasta "*libraries*" podemos descompactá-los, siga as etapas abaixo:

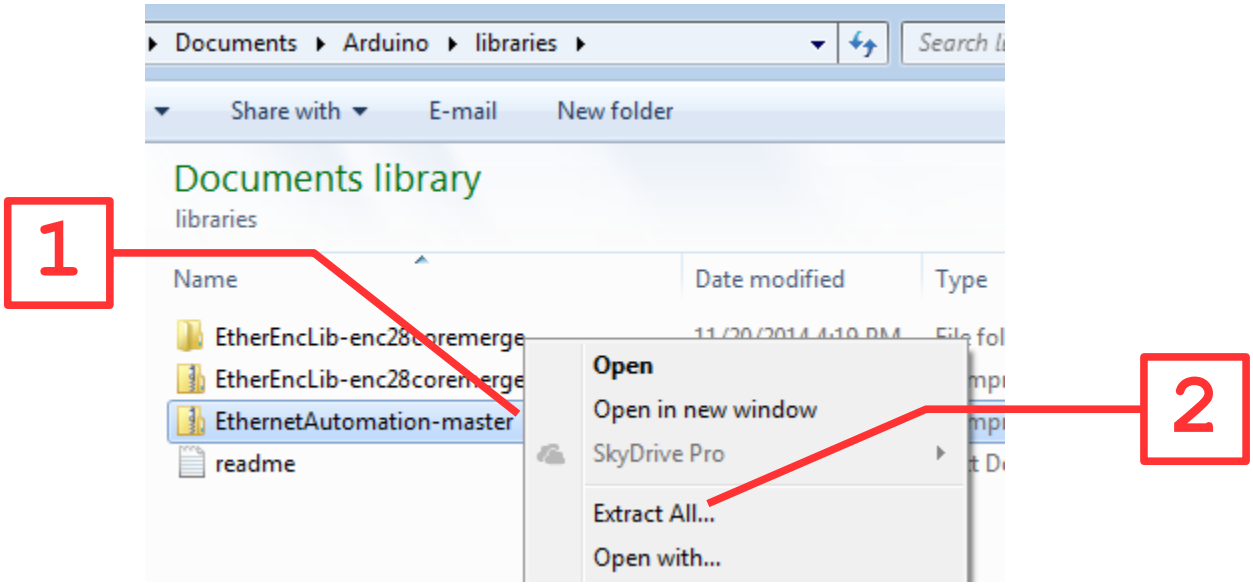


1. Clique com o botão direito em um dos arquivos. O procedimento é o mesmo para ambos, vamos começar pelo **`EtherEncLib-enc28coremerge.zip`**
2. No menu de contexto que aparecer, clique na opção "*Extract All...*" (Extrair tudo...)

Passo 12 – Na tela que aparecer, apenas clique no botão "*Extract*" (Extrair)

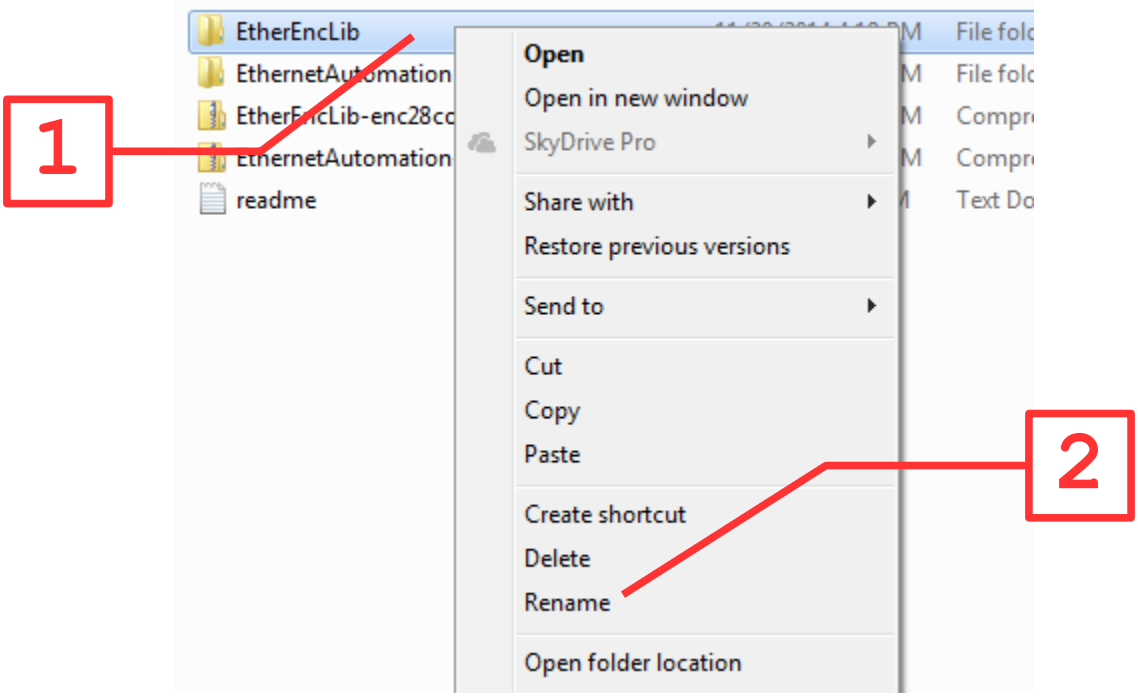


Passo 13 – Repita os mesmos procedimentos dos *Passos 11* e *12* para o arquivo **`EthernetAutomation-master.zip`**



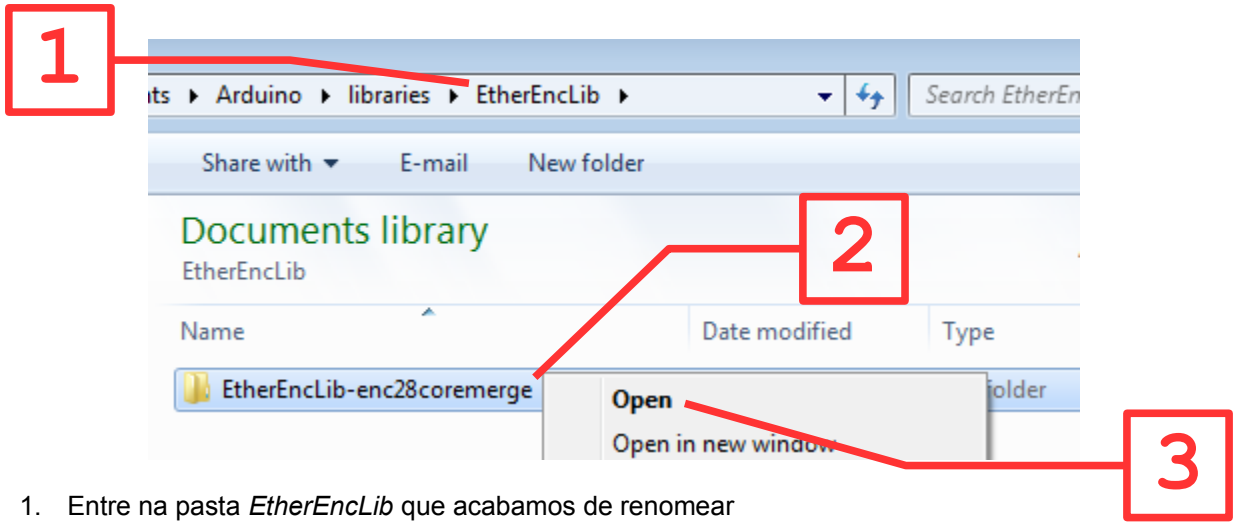
1. Mesmo procedimento, clicando com o botão direito do mouse
2. No menu que aparecer, clique em “Extract All...” (Extrair tudo)

Passo 14 – Agora precisamos renomear a pasta para que funcione no *Arduino*. O *GitHub* adiciona informações no nome da pasta que confundem o *Arduino IDE*.



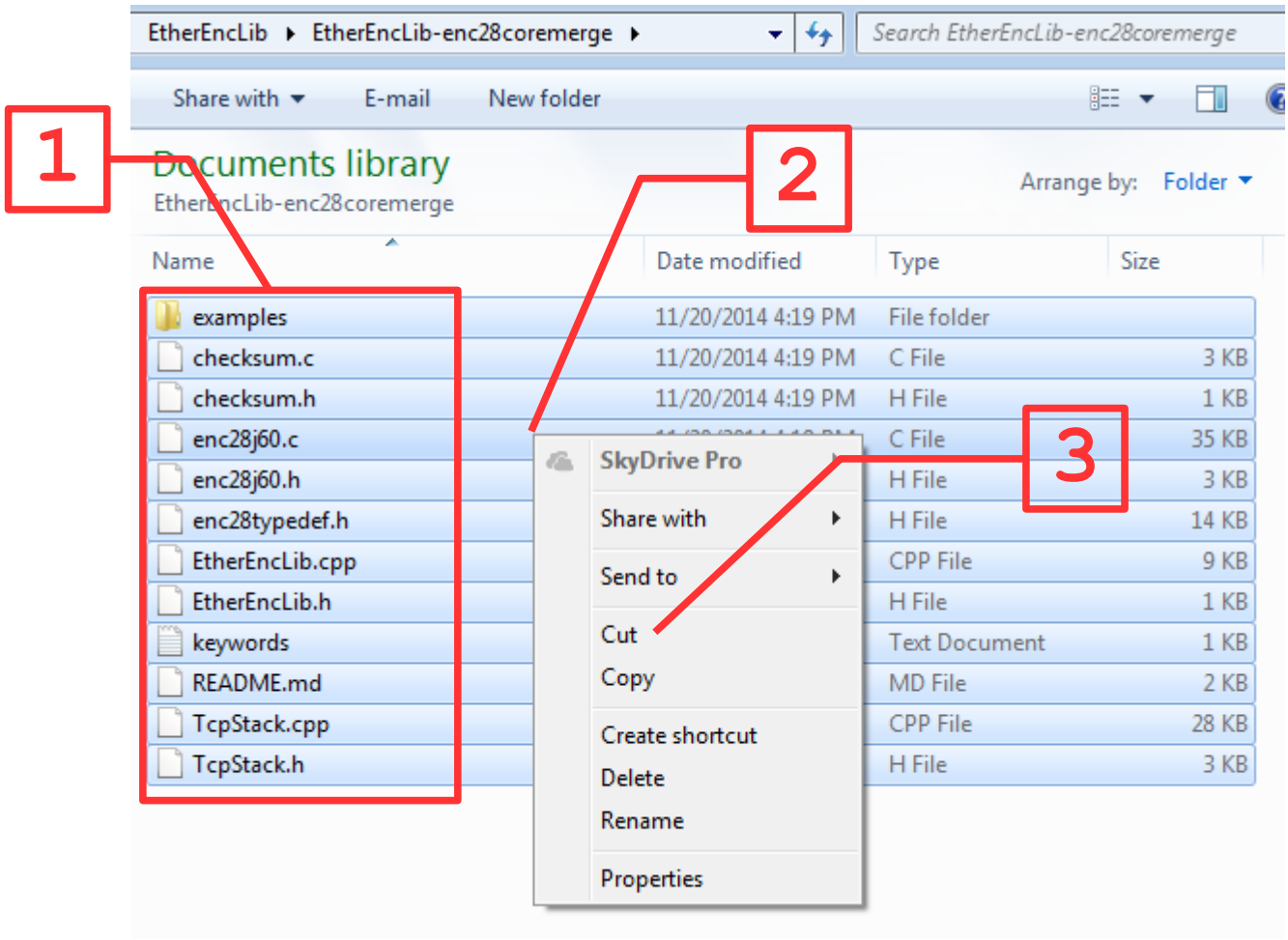
1. Clique com o botão direito do mouse na pasta **`EtherEncLib-enc28coremerge`**
2. No menu que aparecer, clique na opção “*Rename*” (Renomear) e mude o nome para **`EtherEncLib`**, conforme mostrado na figura acima.

Passo 15 – Vamos arrumar os arquivos agora, entrando na pasta que acabamos de renomear. Repare que ao descompactar, criamos uma pasta dentro da outra.



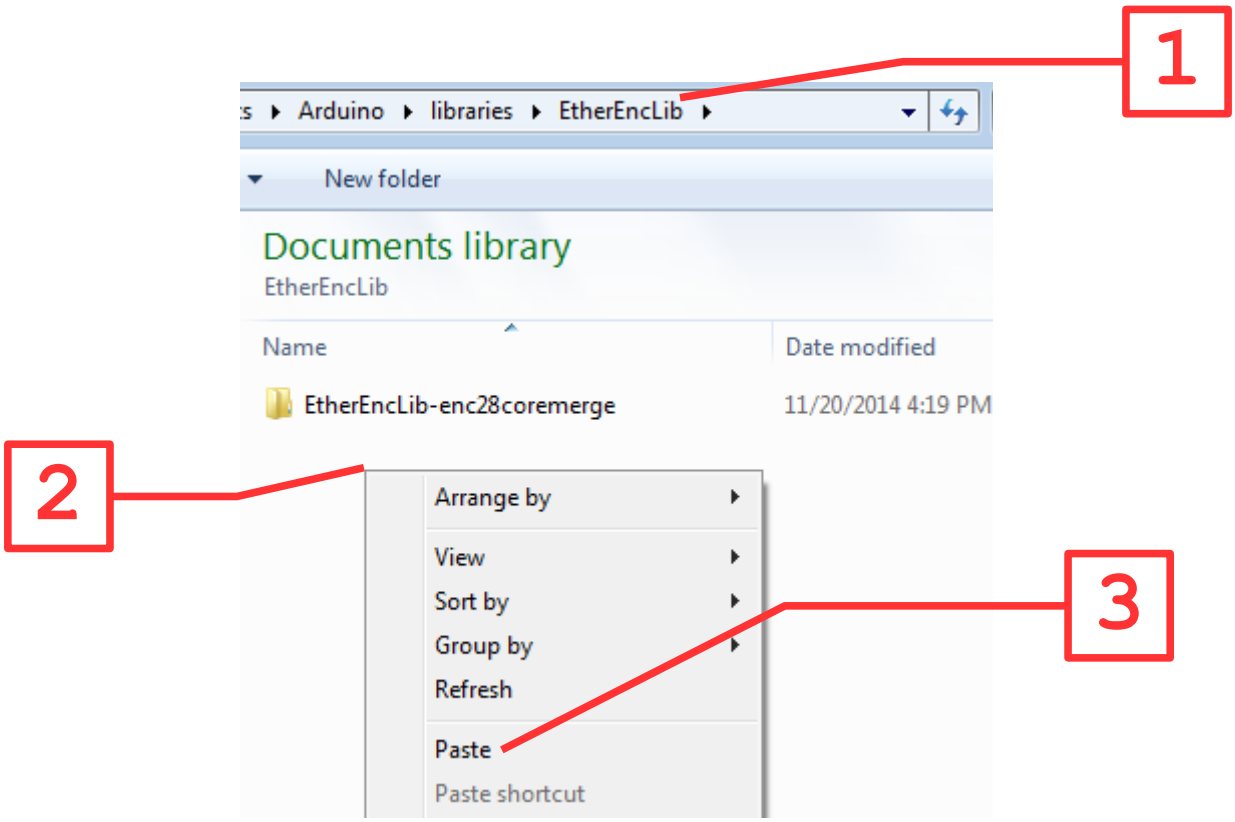
1. Entre na pasta *EtherEncLib* que acabamos de renomear
2. Clique com o botão direito do mouse na sub-pasta (ou simplesmente clique 2x)
3. Selecione a opção "*Open*" (Abrir)

Passo 16 – Vamos selecionar todos os arquivos para movê-los para fora dessa pasta, siga as etapas abaixo:



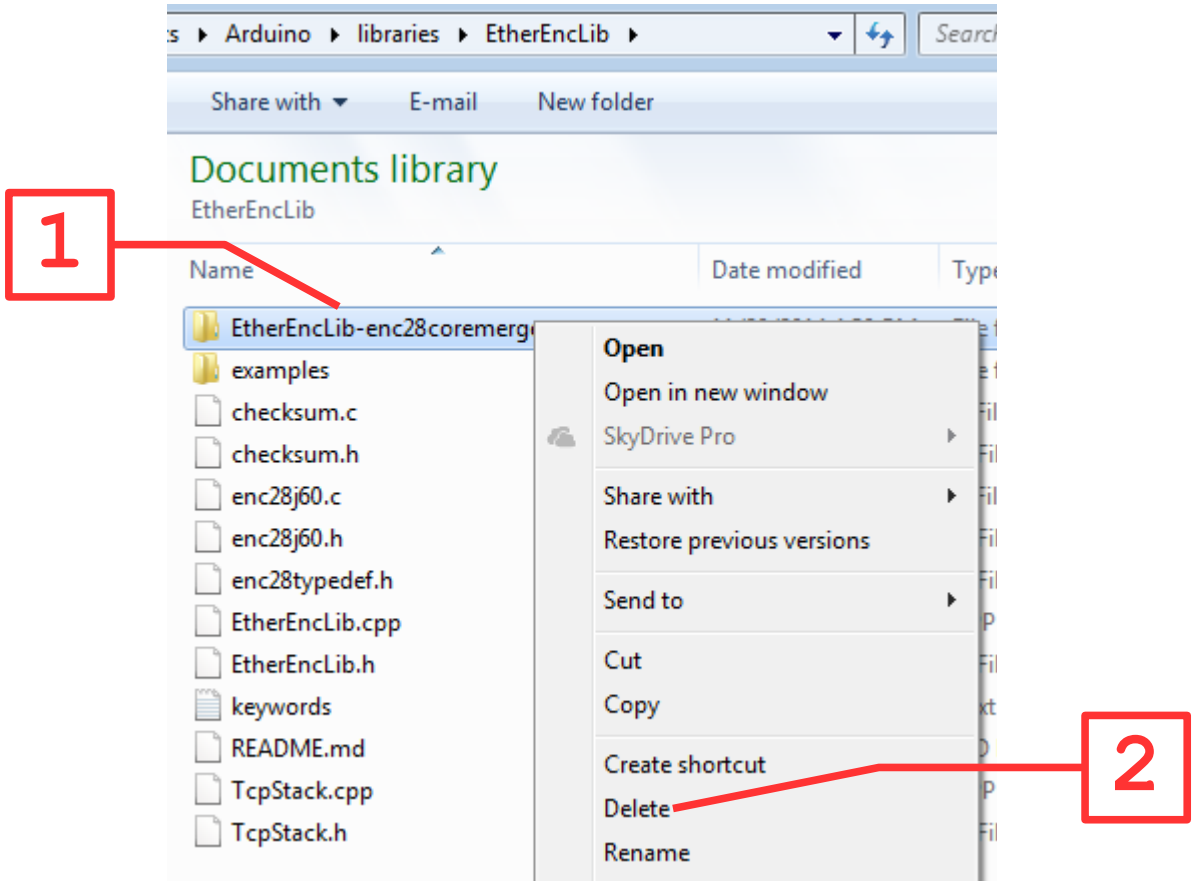
1. Clique e arraste o mouse para selecionar todos os arquivos e diretórios ao mesmo tempo.
2. Clique com o botão direito do mouse para abrir o menu de contexto
3. No menu, clique na opção "*Cut*" (Recortar)

Passo 17 – Voltando na pasta **`EtherEncLib`**, basta colar todos os arquivos recortados.



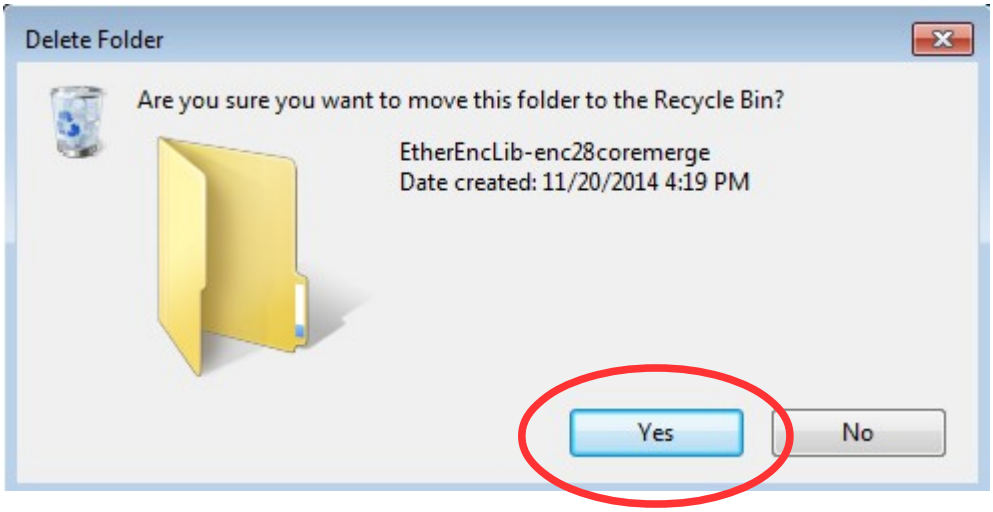
1. Clique na pasta **`EtherEncLib`** para voltar nela
2. Arrume um espaço vazio para clicar com o botão direito do mouse
3. No menu de contexto que aparecer, selecione, clicando na opção “Paste” (Colar)

Passo 18 – Podemos agora apagar a pasta **`EtherEncLib-enc28coremerge`**, pois não vamos precisar dela.

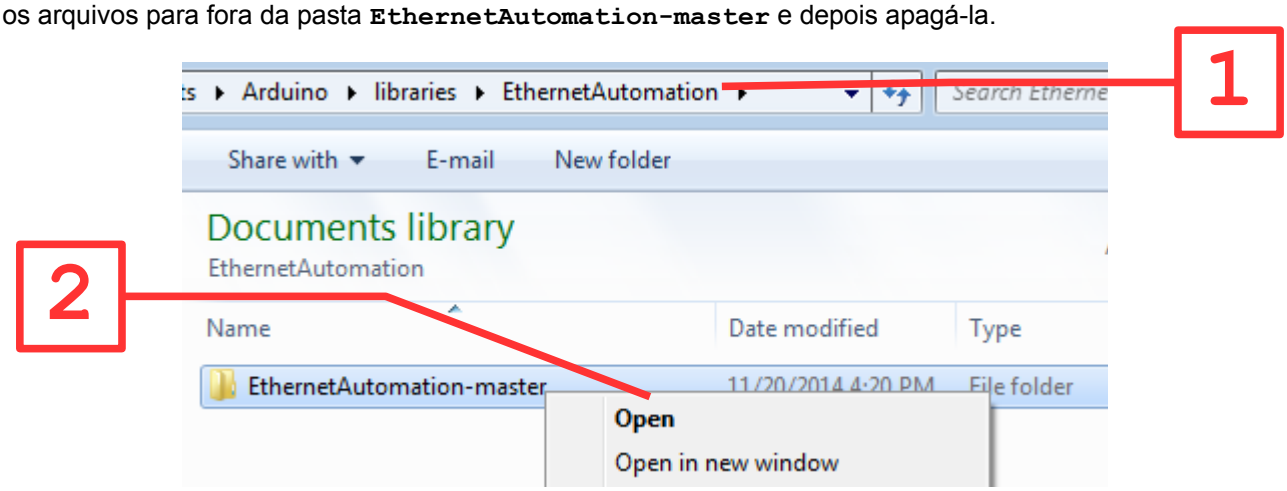


1. Clique com o botão direito na pasta **`EtherEncLib-enc28coremerge`** para aparecer o menu de contexto.
2. Selecione a opção "*Delete*" (Apagar), clicando nela.

Passo 19 – Confirme a operação, clicando no botão "*Yes*", na tela que aparecer.

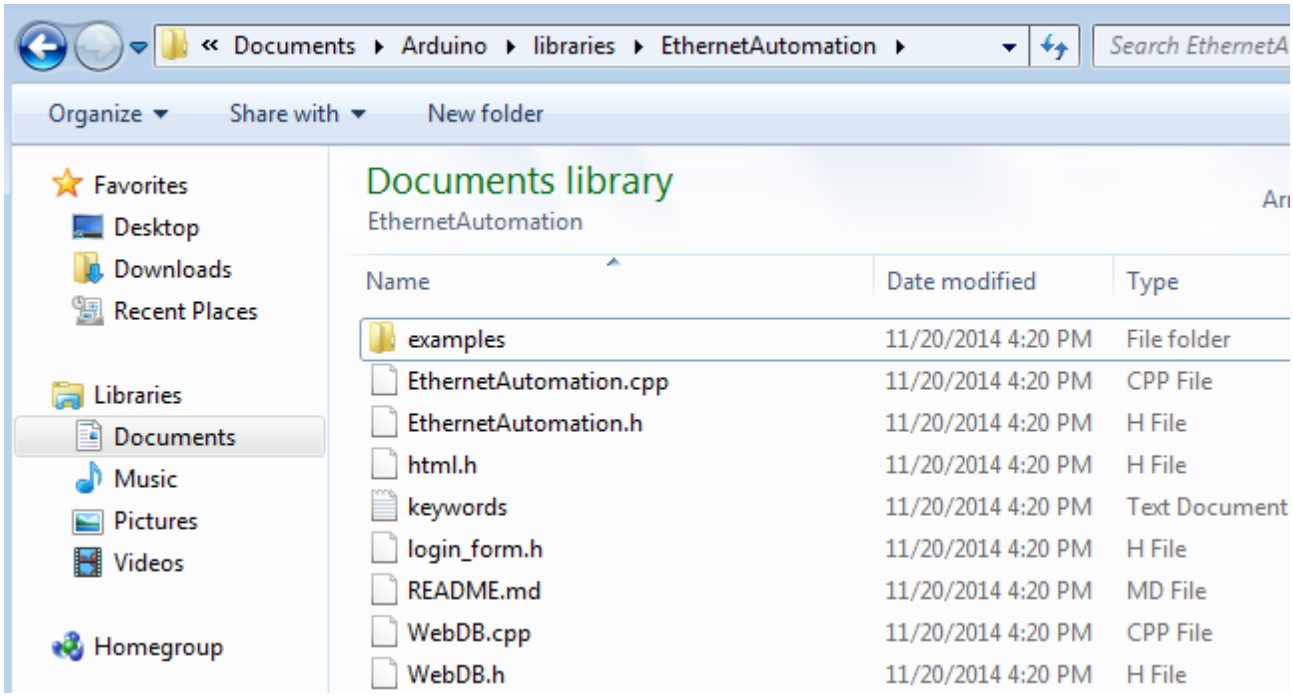


Passo 20 – Repetir os procedimentos dos *Passos 17*, *18* e *19* para a pasta **`EthernetAutomation`**. Mover os arquivos para fora da pasta **`EthernetAutomation-master`** e depois apagá-la.

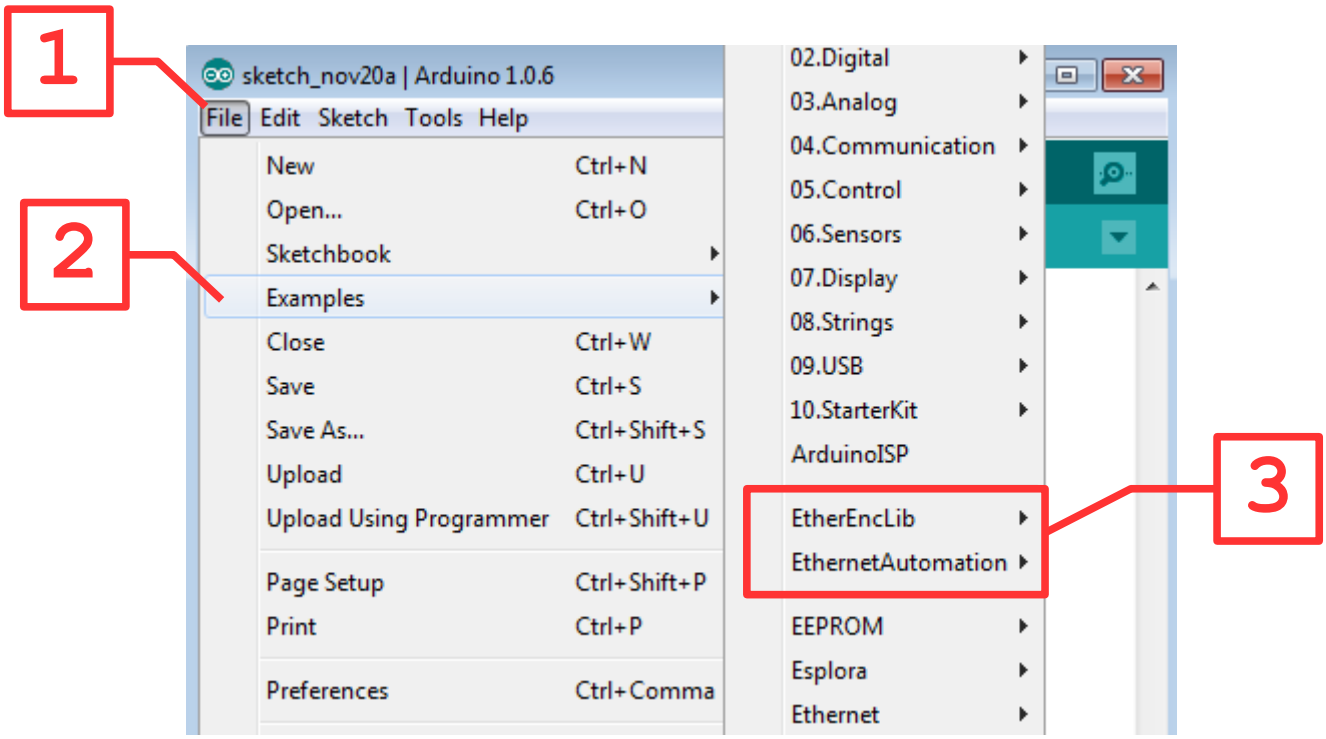


1. Volte na pasta **`EthernetAutomation`**, clicando nela na barra de endereços
2. Abra a pasta e repita os Passos 17, 18 e 19, só que agora para a pasta **`EthernetAutomation-master`**

Passo 21 – O resultado final da pasta **`EthernetAutomation`** deve ficar igual ao mostrado na imagem abaixo:

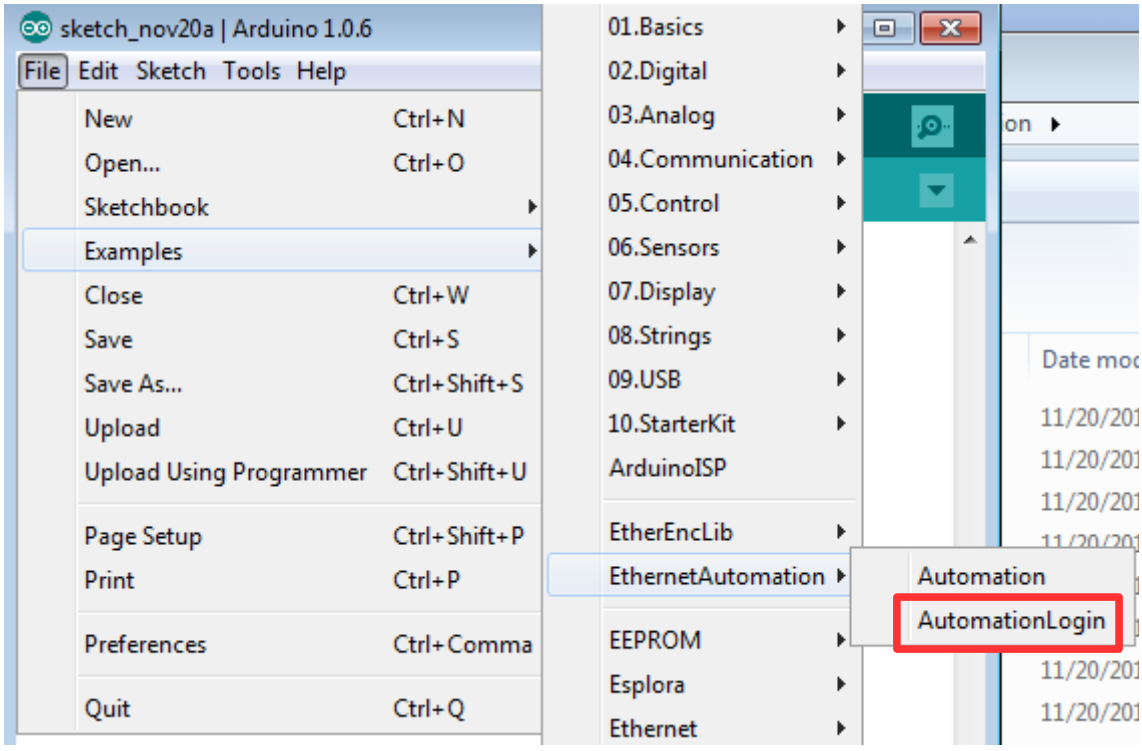


Passo 22 – Agora podemos abrir o *Arduino IDE* e seguir as etapas abaixo:

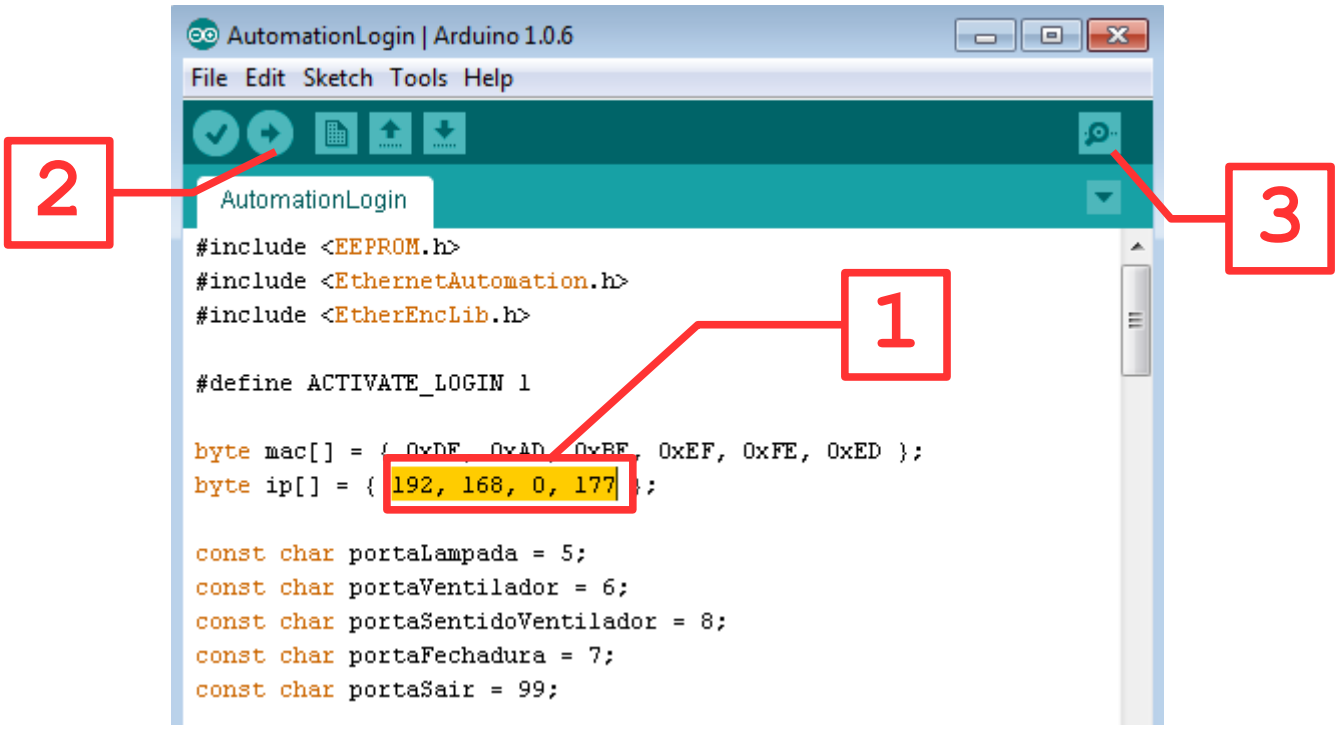


1. Clique no menu "*File*" (Arquivo)
2. Passe o cursor do mouse sobre a opção "*Examples*" (Exemplos)
3. Repare que apareceram as bibliotecas novas que acabamos de instalar!

Passo 23 – Selecione o *Exemplo* chamado **`AutomationLogin`**, conforme mostrado na imagem abaixo:

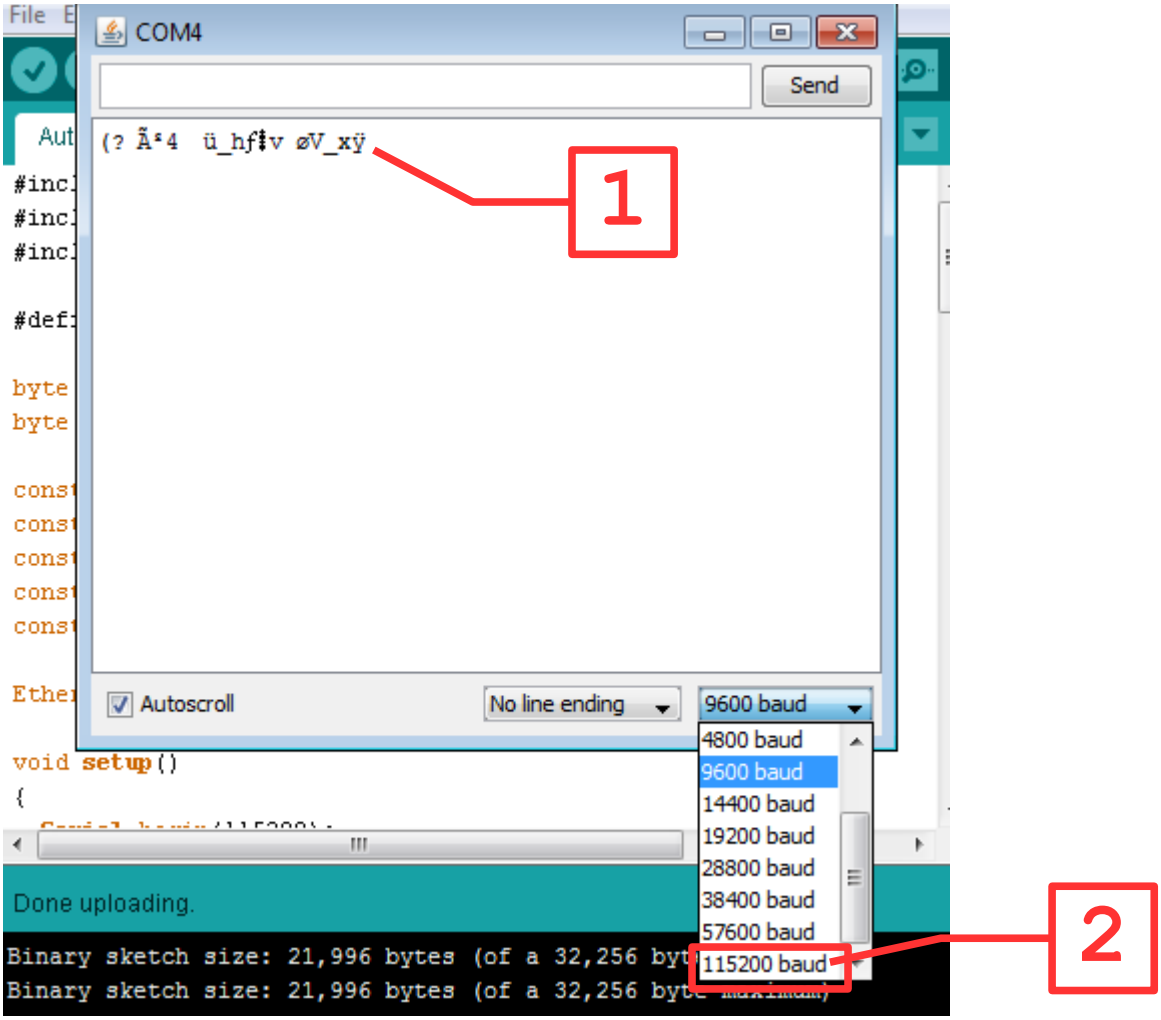


Passo 24 – Configuração básica do código de exemplo com ***Sistema de Logins Automation***. Veremos mais opções de configuração depois, neste manual.



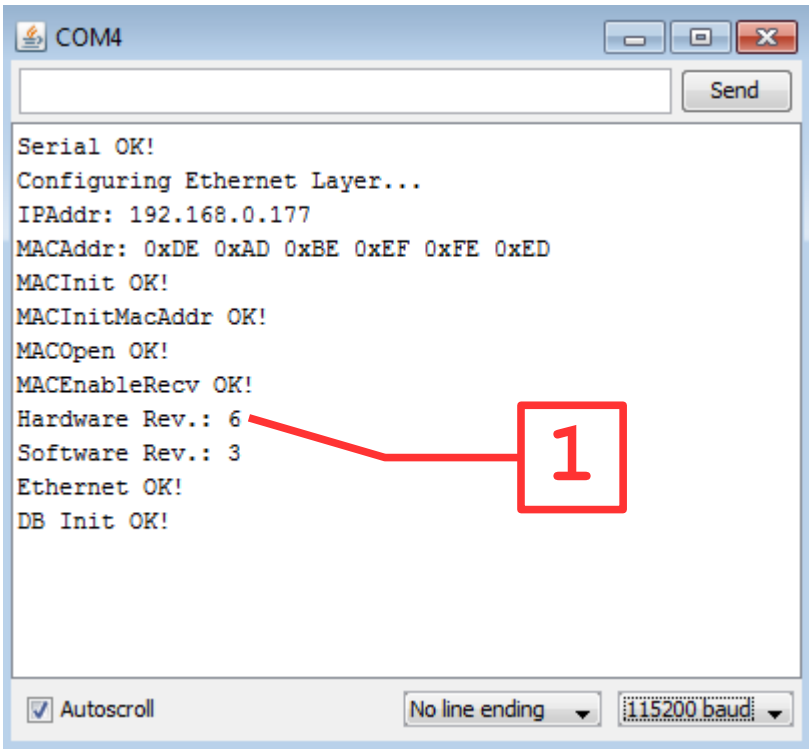
1. Configure um *Endereço IP* válido dentro da sua *Rede de Computadores* (veremos mais detalhes sobre isso, mais adiante).
2. Carregue o código-fonte no *Arduino Supervisório #1*, clicando no ícone mostrado na imagem acima;
3. E abra o *Monitor da Serial* para acompanhar o funcionamento inicial do hardware, clicando no ícone indicado.

Passo 25 – Com o *Monitor da Serial* aberto, precisamos configurar a velocidade (*Baud Rate*) da comunicação, siga as etapas abaixo:



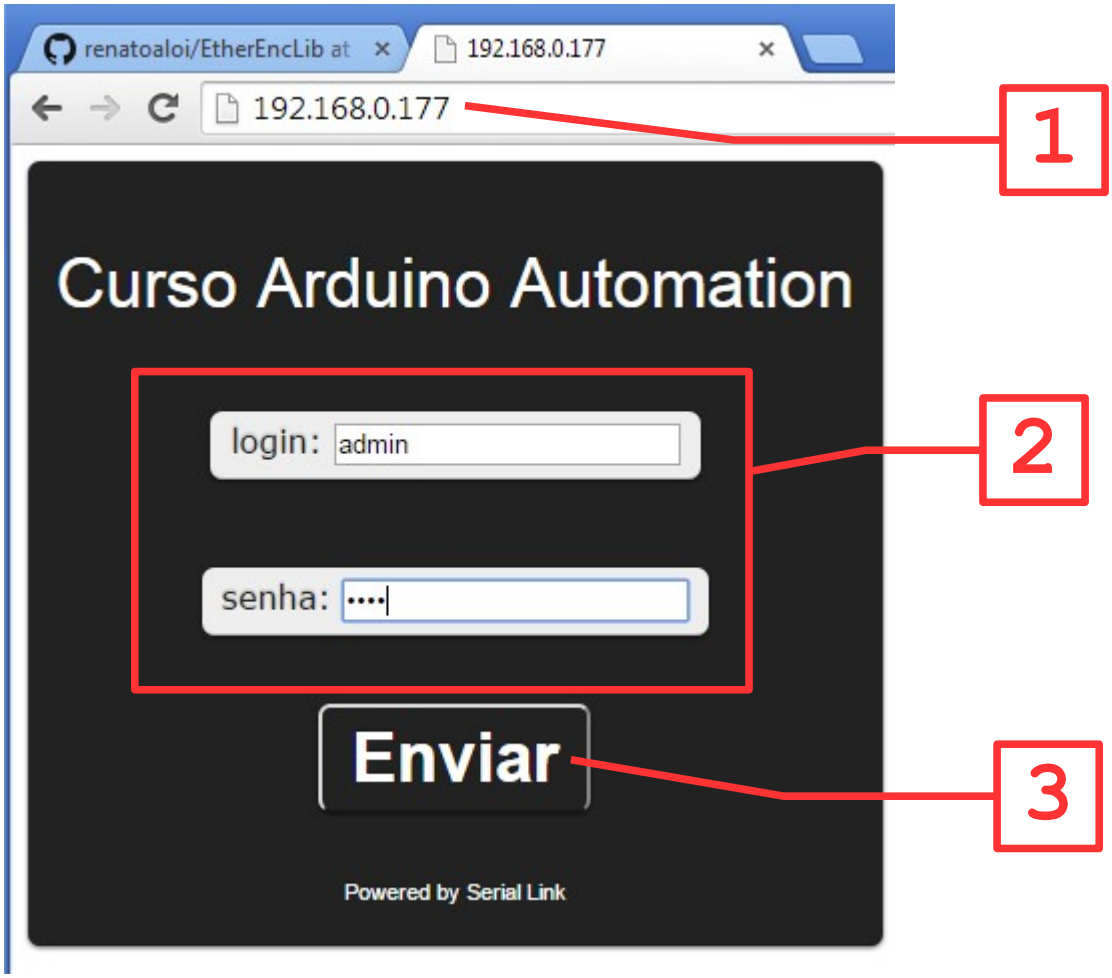
1. Repare que aparecerão caracteres estranhos na *Serial*. Normalmente o *Monitor da Serial* do *Arduino IDE* vem configurado com a velocidade 9600 Bps.
2. Basta configurar para 115200 Bps, conforme mostrado na imagem acima.

Passo 26 – Quando mudarmos a configuração de velocidade da *Comunicação Serial*, o *Arduino Supervisório #1* reiniciará automaticamente e as informações iniciais aparecerão no *Monitor da Serial*.



1. Preste muita atenção para esse número mostrado na imagem acima! Onde está escrito "**`Hardware Rev.: 6`**". Se aparecer "**`Hardware Rev.: 0`**", significa que não funcionou! Reveja os passos e consulte a seção de *Problemas Conhecidos* deste manual. Visite também o *Fórum da Serial Link* e participe da comunidade para aprender e compartilhar suas experiências!

Passo 27 – Com tudo funcionando, podemos voltar no *Navegador* (Browser) e digitar o *Endereço IP* que configuramos para nosso *Painel de Controle Web* do *Kit Automation*, no meu caso **`192.168.0.177`**, conforme segue:



1. Na barra de endereços do *Navegador*, digite o *Endereço IP* que configuramos na código **`AutomationLogin`**
2. Preencha os campos do formulário de login, com os dados abaixo:
    1. login: **`admin`**
    2. senha: **`1234`**
3. Clique no botão “Enviar”

Passo 28 – Pronto! Essa é a tela do ***Painel de Controle Web*** da ***Central de Controle Automation***! Veremos mais detalhes sobre a utilização, depois.

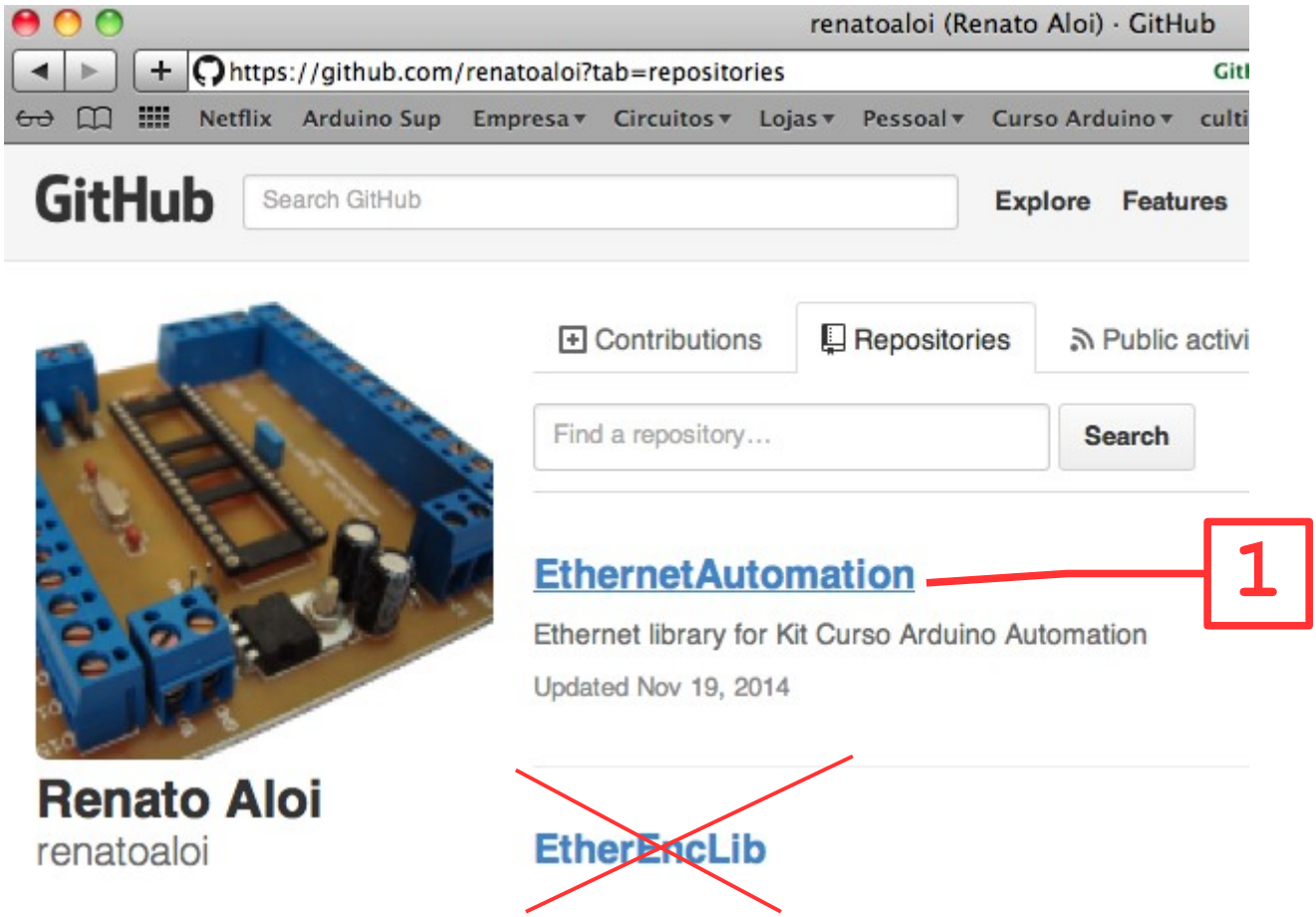
Instalando Bibliotecas Automation no MAC OSX

Vamos ver as instalações agora das bibliotecas ***EthernetAutomation*** e ***EtherEncLib*** no ambiente *MAC OSX*.

Passo 1 – Abrindo site do *GitHub* e acessando o repositório de Renato Aloi.

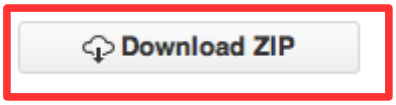
1. Abra seu *Navegador de Internet* preferido e acesse o link: https://github.com/renatoaloi
2. Na página que abrir, clique na aba chamada "*Repositories*" (Repositórios)

Passo 2 – Baixando o *Arquivo ZIP* da biblioteca **`EthernetAutomation`**

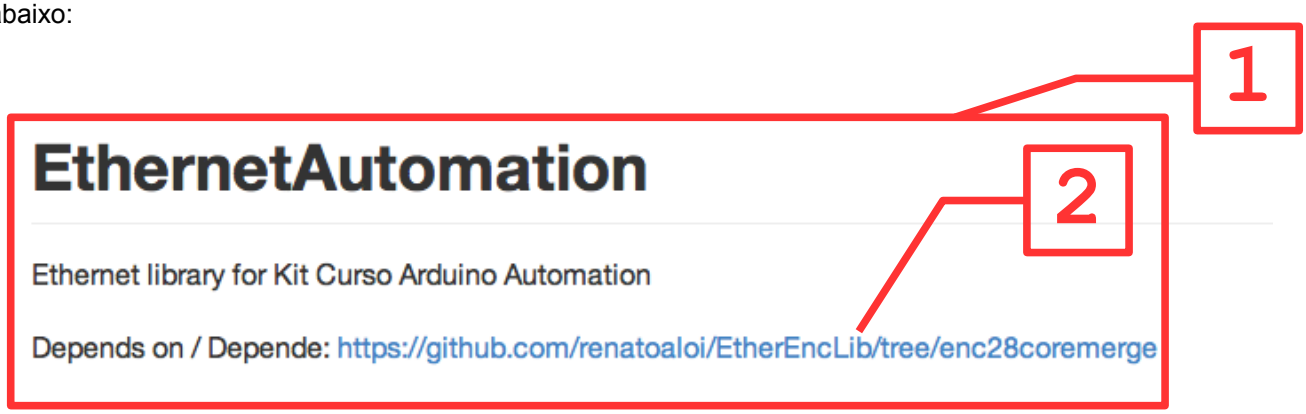


1. Clique no link mostrado na imagem, referente a biblioteca **`EthernetAutomation`**

Passo 3 – Na página que aparecer, procure pelo botão mostrado na imagem abaixo, para fazer o *Download* do *Arquivo ZIP*, e clique nele!



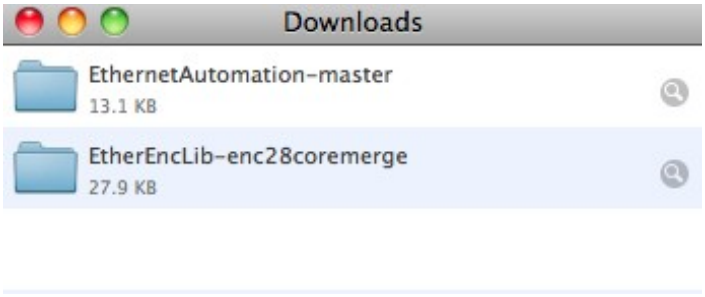
Passo 4 – Procure também, na mesma página, rolando um pouco a tela, por este link mostrado na imagem abaixo:



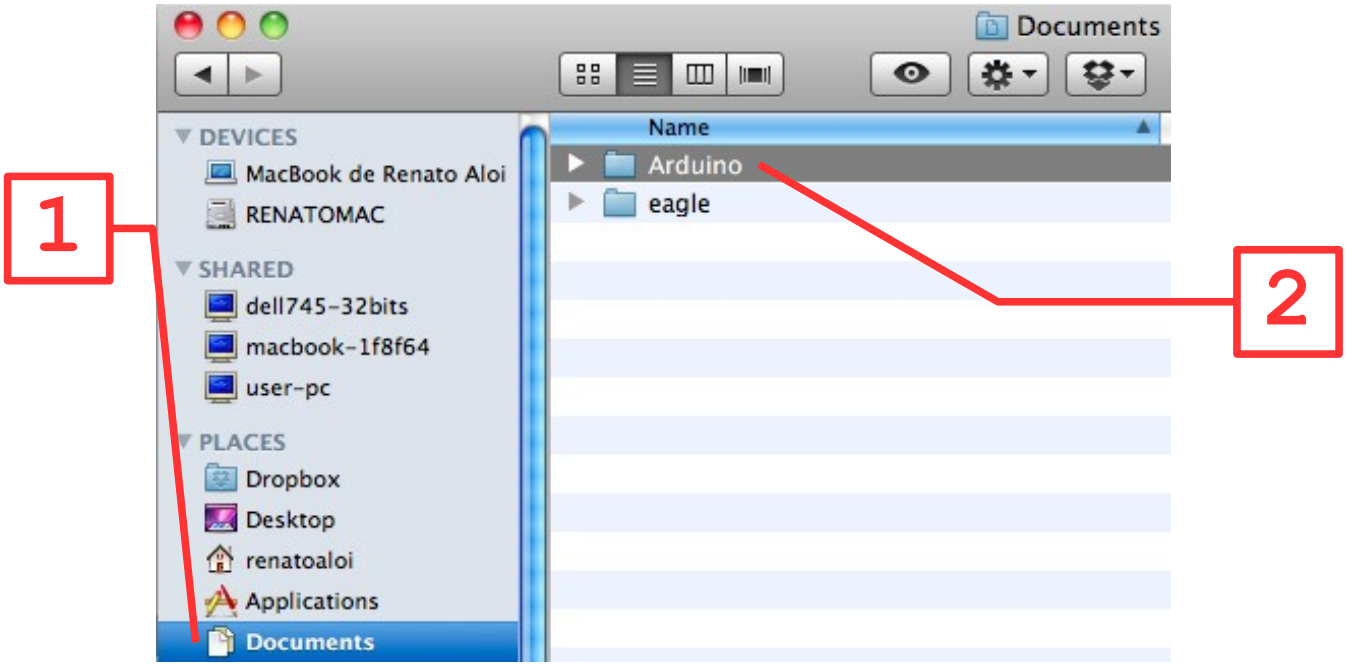
1. Role a tela para achar o texto do *README*, conforme mostrado na imagem
2. Clique no link: https://github.com/renatoaloi/EtherEncLib/tree/enc28coremerge

Passo 5 – Na página que aparecer, da biblioteca **`EtherEncLib`** (ramo: **`enc28coremerge`**), clique no botão *Download ZIP* para iniciar o processo!

Passo 6 – No *MAC*, as pastas compactadas são automaticamente convertidas em pastas normais. Então facilita muito o trabalho. Veja como ficaram os *Downloads* que acabamos de fazer.

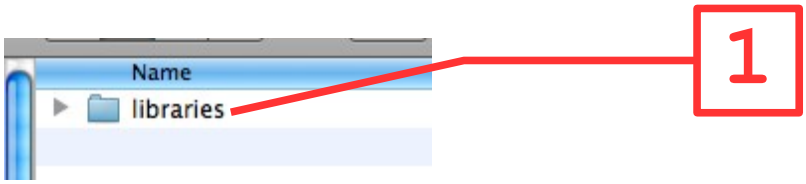


Passo 7 – Abra a pasta de *Downloads*, do passo anterior. Mas também navegue até a pasta dos nossos *Sketches* do *Arduino*.



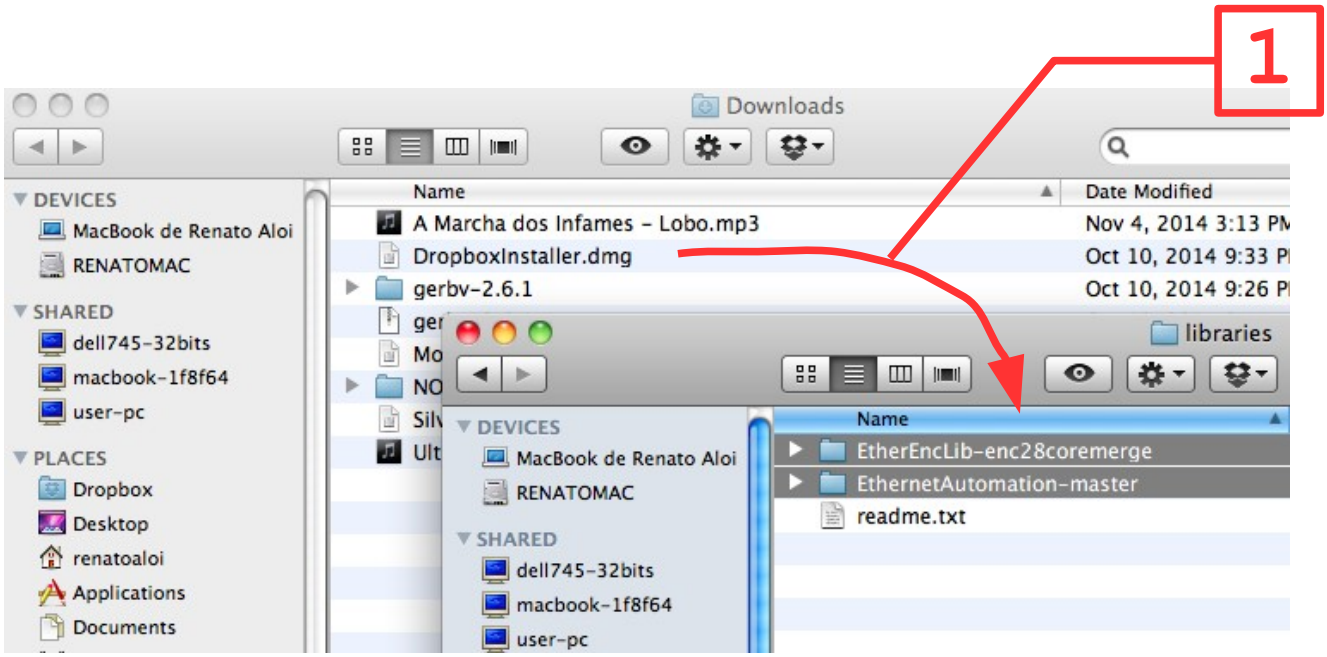
1. Abra o "*Finder*" e clique no atalho para "*Documents*" (Documentos)
2. Dentro dessa pasta temos que clicar na pasta *Arduino* 2x, para abrí-la.

Passo 8 – Dentro da pasta dos *Sketches* do *Arduino*, temos a pasta "*libraries*". Esta pasta é onde instalaremos nossa bibliotecas!



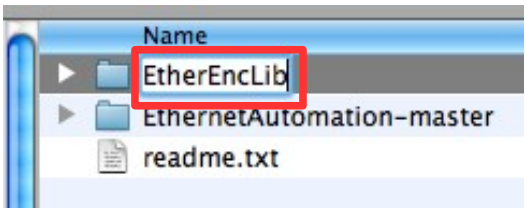
1. Clique 2x também na pasta "*libraries*" para abri-la!

Passo 9 – Arrastando pastas das bibliotecas **`EthernetAutomation`** e **`EtherEncLib`**, da pasta *Downloads* para a pasta correta, chamada "*libraries*", que acabamos de navegar.

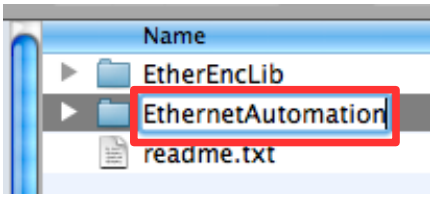


1. Arraste as duas pastas **`EthernetAutomation`** e **`EtherEncLib`** da pasta de *Downloads* para a pasta "*libraries*", dentro da pasta dos *Sketches* do *Arduino*.

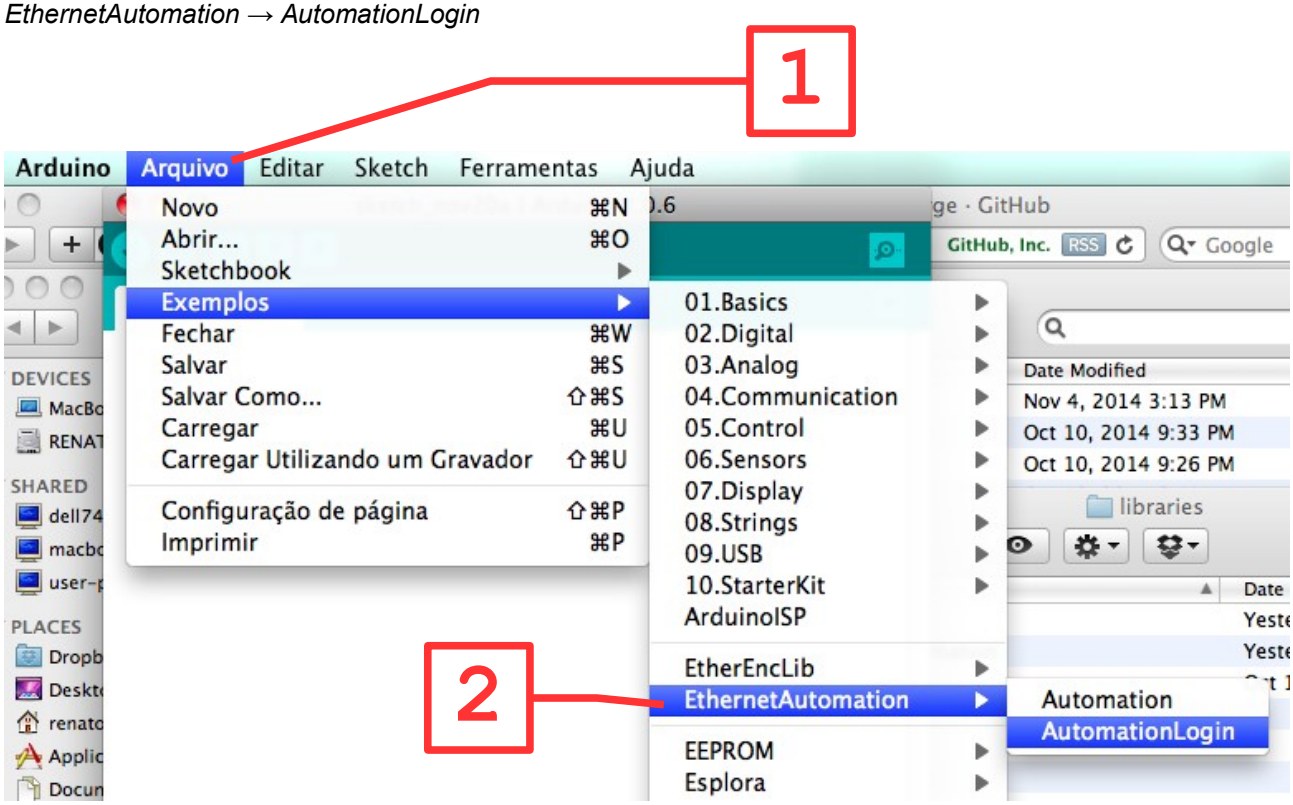
Passo 10 – Finalmente, precisamos renomear as pastas **`EtherEncLib-enc28coremerge`** para **`EtherEncLib`**



Passo 11 – E precisamos renomear as pastas **`EthernetAutomation-master`** para **`EthernetAutomation`**

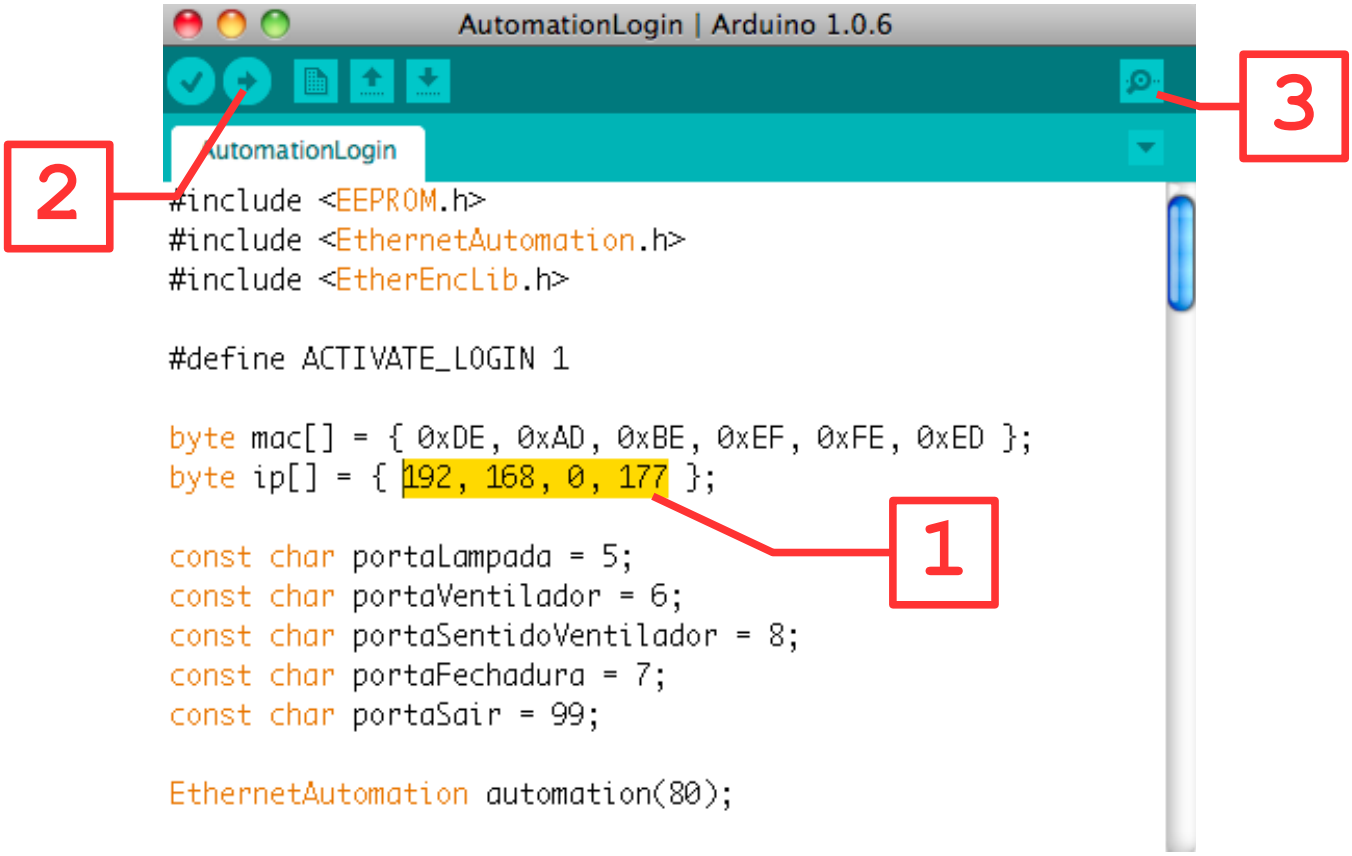


Passo 12 – Neste ponto, basta abrir o *Arduino IDE* e verificar nos menus *Arquivo* → *Exemplos* → *EthernetAutomation* → *AutomationLogin*



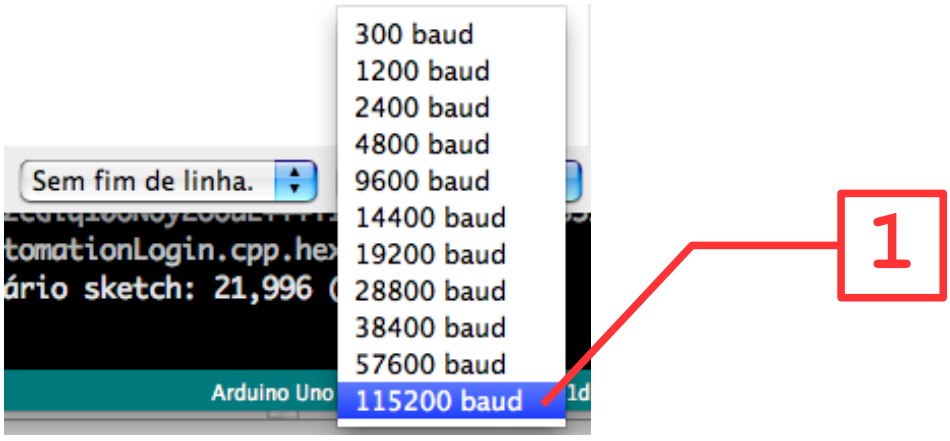
1. Abra o *Arduino IDE* e navegue pelos menus *Arquivo* → *Exemplos*
2. Verifique que as novas bibliotecas apareceram. Selecione o exemplo **`AutomationLogin`** e vamos editá-lo.

Passo 13 – Vamos apenas configurar o *Endereço IP*, modificando a parte marcada, no código mostrado abaixo. Utilize um IP compatível com sua *Rede de Computadores*. Veremos mais detalhes dessa parte na seção de *Utilização do Kit Automation*, mais adiante, neste manual.



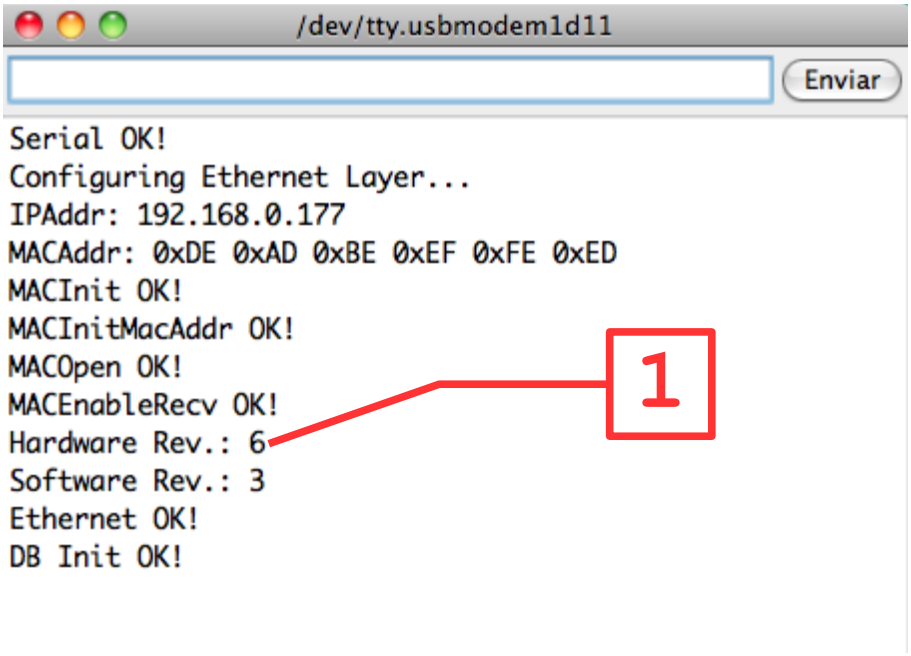
1. Modifique o *Endereço IP* para ser compatível com sua *Rede de Computadores*
2. Clique no ícone marcado na imagem para carregar o código-fonte para o *Arduino*
3. Quando terminar o carregamento, clique no ícone do *Monitor da Serial*, mostrado na imagem.

Passo 14 – Ajuste da velocidade da *Comunicação Serial*.



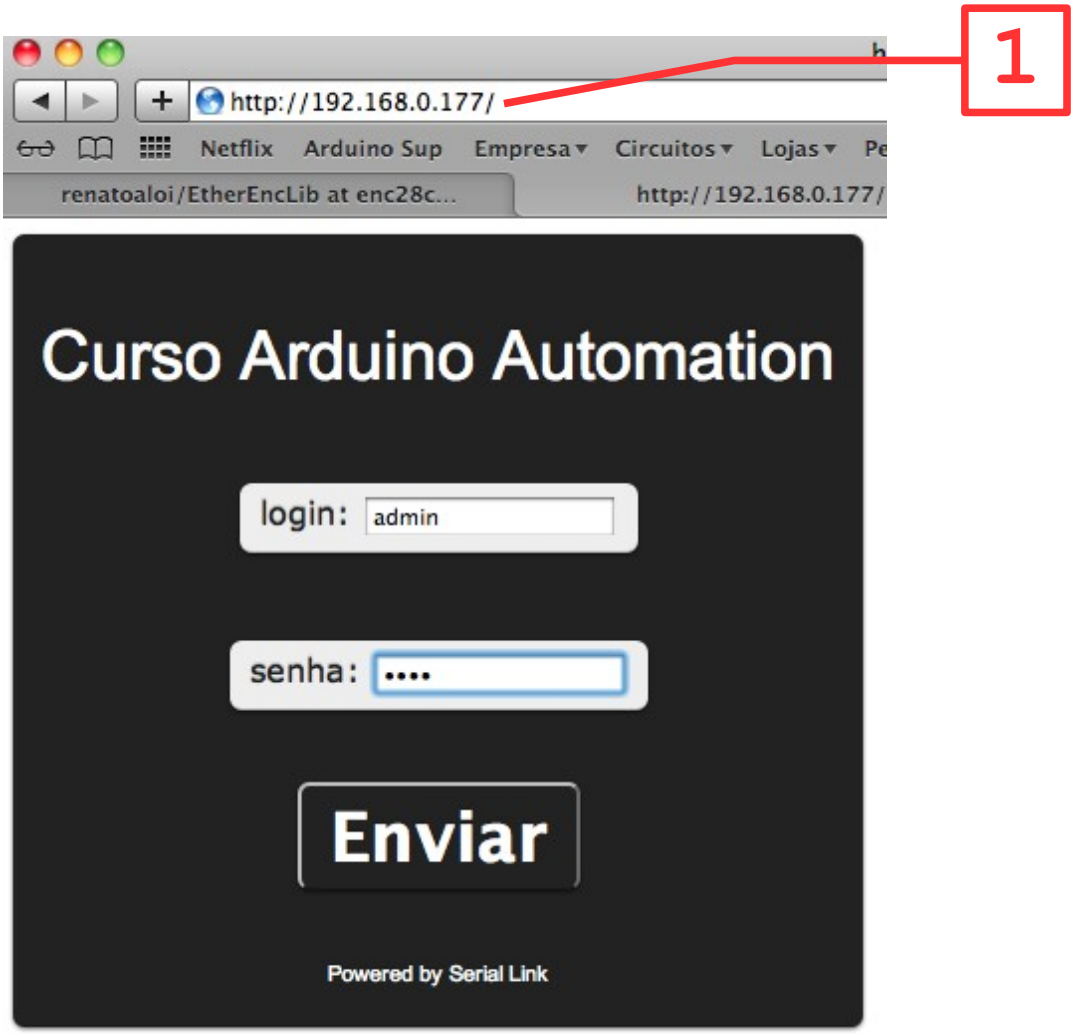
1. Ao abrir o *Monitor da Serial*, regule a velocidade para 115200 Bps, conforme mostrado acima. Não se preocupe se caracteres estranhos aparecerem na *Serial*. É normal.

Passo 15 – Com a velocidade da *Comunicação Serial* ajustada para 115200 Bps, as informações corretas começam a aparecer. Siga a etapa abaixo para verificar se tudo está funcionando corretamente.



1. Verifique se o dígito `6` (seis) aparece nessa linha, indicada na imagem. Se parecer o dígito `0` (zero) significa que NÃO está funcionando. Verifique isso agora, para fazer os ajustes necessários. Verifique a seção de *Utilização do Kit Automation* e *Problemas Conhecidos*, para maiores informações.

Passo 16 – Volte ao *Navegador de Internet* para acessar o *Endereço IP* que configuramos para nosso *Kit Automation*.

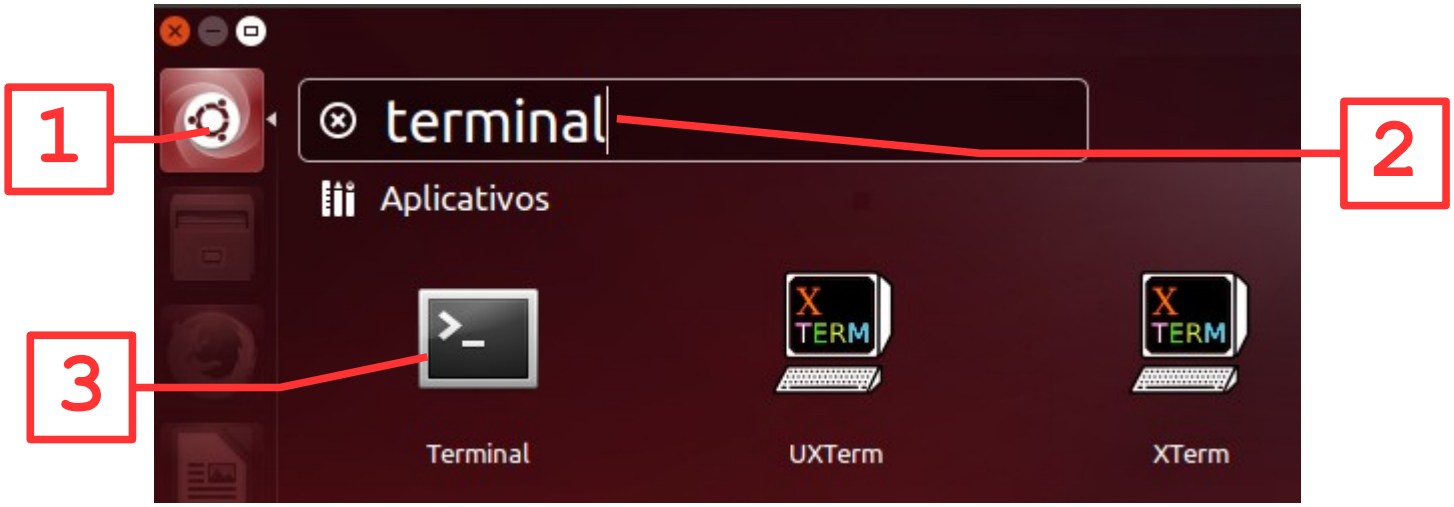


1. Digite o *Endereço IP* do *Kit Automation*, que ajustamos algumas etapas atrás. Estamos prontos! Siga os passos da parte de *Utilização do Kit Automation*, mais adiante, neste manual.

Instalando Bibliotecas Automation no Linux

Vamos ver as instalações agora das bibliotecas ***EthernetAutomation*** e ***EtherEncLib*** no ambiente Linux.

Passo 1 – Primeiro vamos abrir o Terminal para digitar os comandos. O jeito mais fácil de instalar as bibliotecas no Linux é através de um programa cliente do Git.



1. Clique no botão indicado na imagem (Ubuntu)
2. Pesquise pelo Terminal Bash
3. Clique no aplicativo “Terminal” para iniciá-lo!

Passo 2 – Executar o comando para instalar o git

```
$ apt-get install git-core gitk
```

```
Preparing to unpack .../git_1%3a1.9.1-1_i386.deb ...
Unpacking git (1:1.9.1-1) ...
A seleccionar pacote anteriormente não seleccionado git-core.
Preparing to unpack .../git-core_1%3a1.9.1-1_all.deb ...
Unpacking git-core (1:1.9.1-1) ...
A seleccionar pacote anteriormente não seleccionado gitk.
Preparing to unpack .../gitk_1%3a1.9.1-1_all.deb ...
Unpacking gitk (1:1.9.1-1) ...
Processing triggers for man-db (2.6.7.1-1) ...
Configurando liberror-perl (0.17-1.1) ...
Configurando git-man (1:1.9.1-1) ...
Configurando git (1:1.9.1-1) ...
Configurando git-core (1:1.9.1-1) ...
Configurando gitk (1:1.9.1-1) ...
cursoarduino@cursoarduino-VirtualBox:~$
```

Passo 3 – Alterando a pasta atual para diretório "*libraries*" do Arduino.

```
$ cd /home/nomeusuario/sketchbook/libraries
```

```
Configurando git (1:1.9.1-1) ...
Configurando git-core (1:1.9.1-1) ...
@cursoarduino-VirtualBox:~$ cd /home/cursoarduino/sketchbook/librari
```

Passo 4 – Execute o comando Git para clonar o repositório da biblioteca **`EthernetAutomation`**

```
$ git clone https://github.com/renatoaloi/EthernetAutomation
```

```
cursoarduino@cursoarduino-VirtualBox:~/sketchbook/libraries$ git clone https://g
ithub.com/renatoaloi/EthernetAutomation
Cloning into 'EthernetAutomation'...
remote: Counting objects: 20, done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 20 (delta 3), reused 17 (delta 2)
Unpacking objects: 100% (20/20), done.
Checking connectivity... done.
cursoarduino@cursoarduino-VirtualBox:~/sketchbook/libraries$
```

Passo 5 – Execute o comando Git também para clonar o repositório da biblioteca **`EtherEncLib`**

```
$ git clone https://github.com/renatoaloi/EtherEncLib
```

```
cursoarduino@cursoarduino-VirtualBox:~/sketchbook/libraries$ git clone https://g
ithub.com/renatoaloi/EtherEncLib
Cloning into 'EtherEncLib'...
remote: Counting objects: 67, done.
remote: Compressing objects: 100% (46/46), done.
remote: Total 67 (delta 22), reused 49 (delta 13)
Unpacking objects: 100% (67/67), done.
Checking connectivity... done.
cursoarduino@cursoarduino-VirtualBox:~/sketchbook/libraries$
```

Passo 6 –Vamos mudar para a pasta **`EtherEncLib`** Execute o comando abaixo:

```
$ cd EtherEncLib
```

```
-rw-rw-r-- 1 cursoarduino cursoarduino  3453 Nov 20 19:40 EtherEncLib.h
drwxrwxr-x 5 cursoarduino cursoarduino  4096 Nov 20 19:40 examples/
drwxrwxr-x 8 cursoarduino cursoarduino  4096 Nov 20 19:40 .git/
-rw-rw-r-- 1 cursoarduino cursoarduino    41 Nov 20 19:40 keywords.txt
-rw-rw-r-- 1 cursoarduino cursoarduino 12746 Nov 20 19:40 net.h
-rw-rw-r-- 1 cursoarduino cursoarduino  1175 Nov 20 19:40 README.md
-rw-rw-r-- 1 cursoarduino cursoarduino  7650 Nov 20 19:40 socket.c
-rw-rw-r-- 1 cursoarduino cursoarduino   877 Nov 20 19:40 socket.h
cursoarduino@cursoarduino-VirtualBox:~/sketchbook/libraries/EtherEncLib$
```

Passo 7 – Precisamos pegar um ramo (branch) específico do repositório. Para isso execute o seguinte comando, dentro da pasta **`EtherEncLib`**:

```
$ git checkout enc28coremerge
```

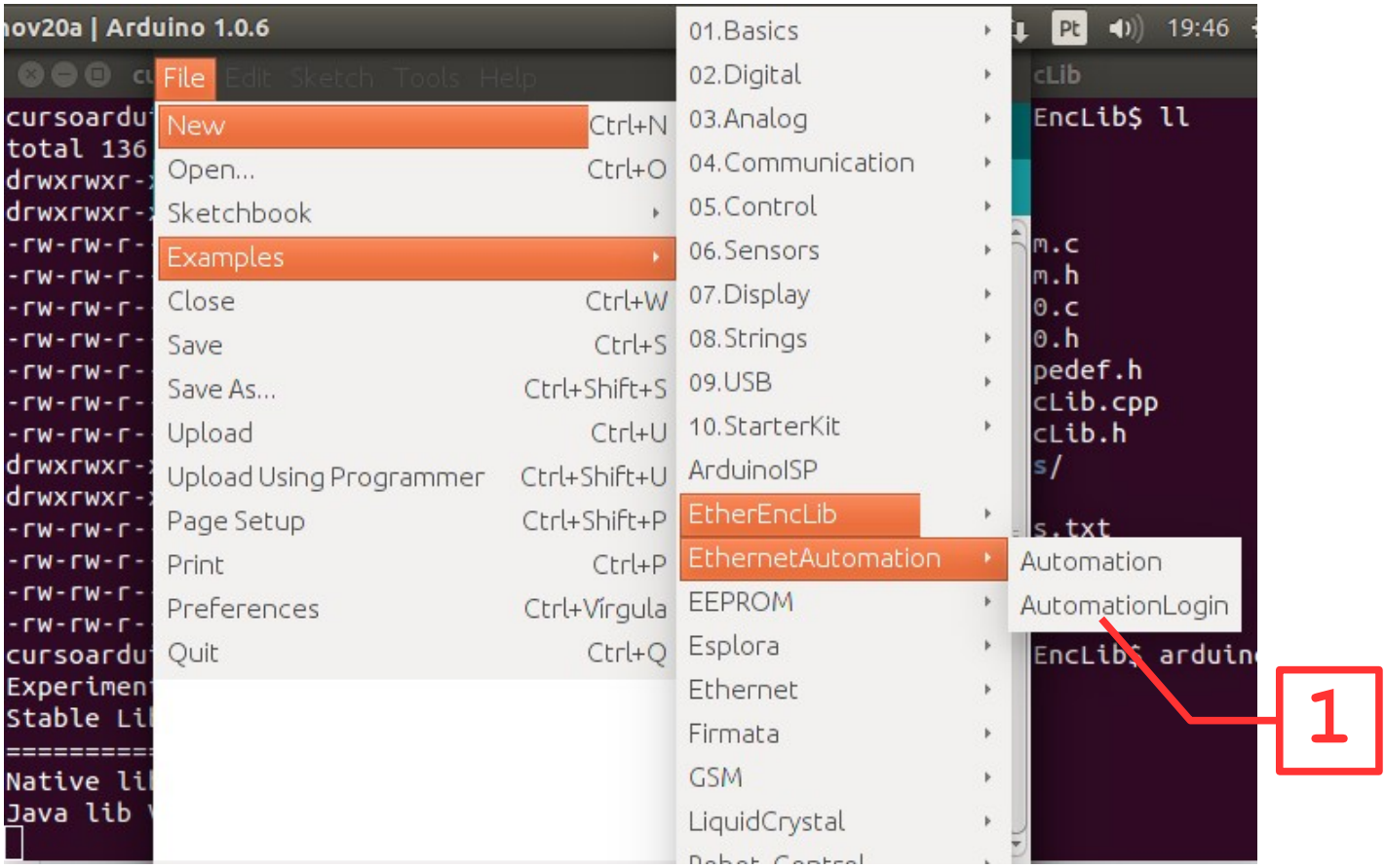
```
cursoarduino@cursoarduino-VirtualBox:~/sketchbook/libraries/EtherEncLib$ git che
ckout enc28coremerge
Branch enc28coremerge set up to track remote branch enc28coremerge from origin.
Switched to a new branch 'enc28coremerge'
cursoarduino@cursoarduino-VirtualBox:~/sketchbook/libraries/EtherEncLib$ git pul
l . enc28coremerge
From .
 * branch            enc28coremerge -> FETCH_HEAD
Already up-to-date.
cursoarduino@cursoarduino-VirtualBox:~/sketchbook/libraries/EtherEncLib$ ll
```

Depois execute o comando abaixo para verificar se está tudo ok, conforme mostrado na imagem acima:

```
$ git pull . enc28coremerge
```

Passo 8 – Tudo OK! Podemos abrir o Arduino IDE, digitando o comando:

```
$ arduino
```



1. Navegue pelos menus File → Examples → EthernetAutomation → AutomationLogin e siga os passos de configuração mostrados na instalação para Windows.

Configurando e Carregando o Código dos Dimmers

Nesta última etapa de instalação, vamos configurar e carregar o código responsável por acionar os dois ***Módulos Dimmer AC*** que acompanham o ***Kit Curso Arduino Automation***.

Agora é a hora de desligar o Arduino FTDi Basic do Arduino Supervisório #1 e ligá-lo no #2. Faça isso antes de começarmos com o passo-a-passo.

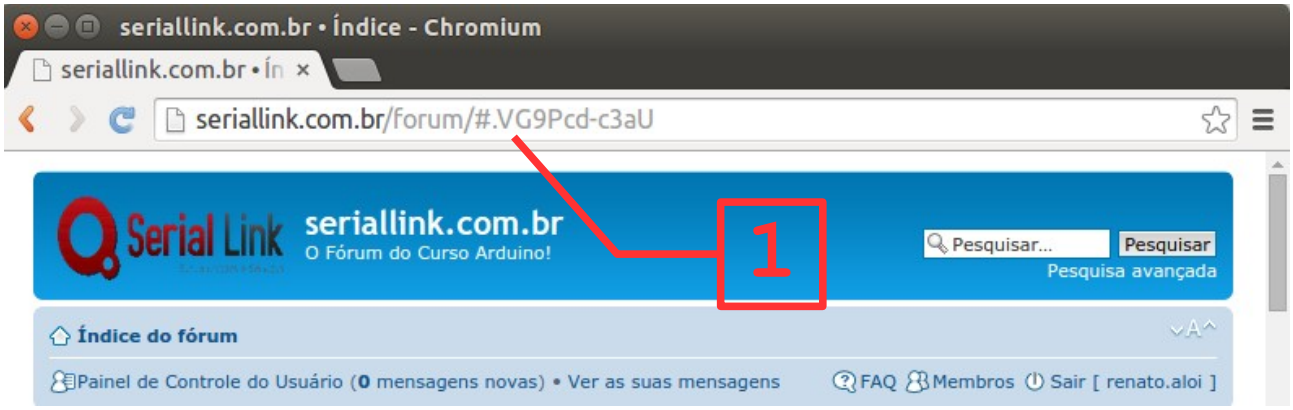
Esta etapa não depende do *Sistema Operacional*, então tanto para usuários do *Windows*, *MAC OSX*, ou *Linux*, a partir de agora, os procedimentos são os mesmos!

Atenção – A instalação das bibliotecas mostradas a seguir considera que você já está com o seu ***Arduino FTDI Basic*** devidamente conectado no ***Arduino Supervisório REV2***. Mas qual? Essa é a parte importante! Essas bibliotecas devem ser carregadas no *Arduino Supervisório* ***#2***, que está conectado no *Módulo Ethernet Enc28J60*.

Instalando Código dos Dimmers AC

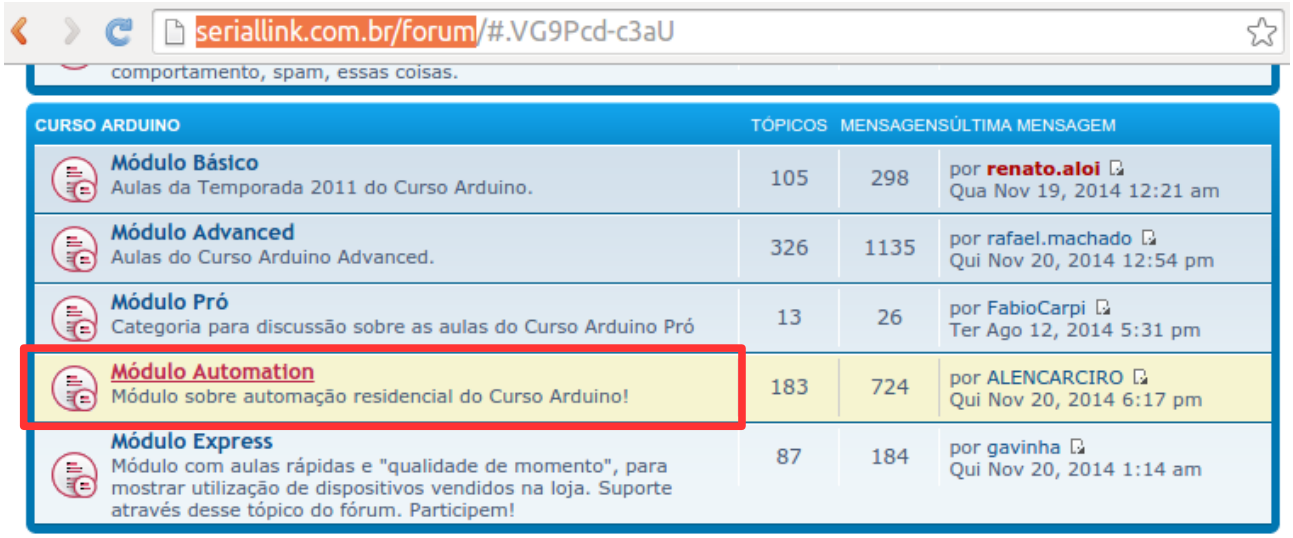
Para correto funcionamento do ***Kit Curso Arduino Automation***, vamos precisar carregar um código no ***Arduino Supervisório #2***. Conecte o ***Arduino FTDi Basic*** no ***Arduino Supervisório #2***, localizado ao lado dos ***Módulos Dimmer AC***, e vamos seguir os passos.

Passo 1 – Acesse o *Fórum da Serial Link* para pegar a versão mais atual do código-fonte.

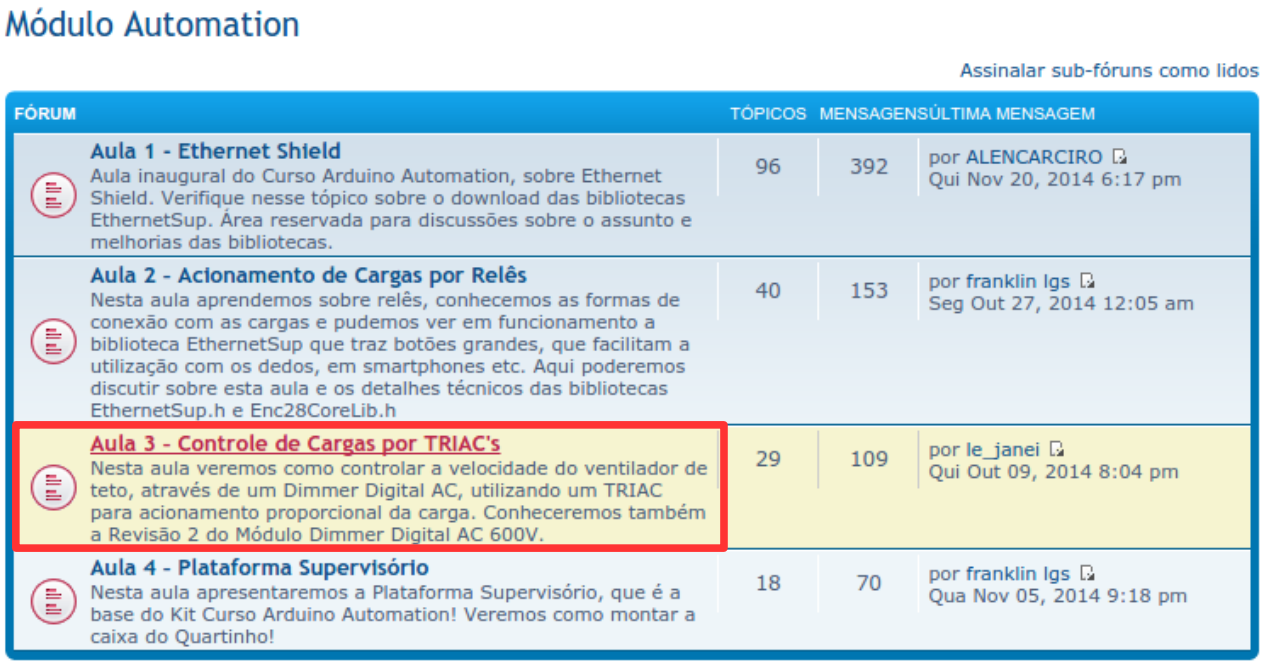


1. Navegue até o endereço do Fórum da Serial Link: http://seriallink.com.br/forum

Passo 2 – Role a página e procure pelo *Tópico do Módulo Automation* do *Curso Arduino*, conforme imagem abaixo:

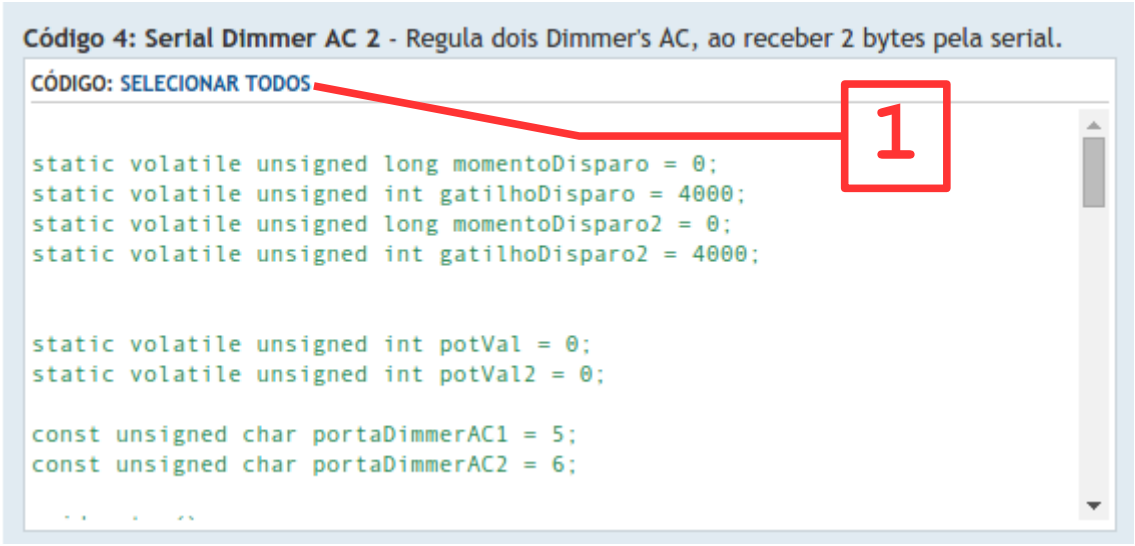


Passo 3 – Na página que abrir, localize e clique no *Tópico da Aula 3 – Controle de Cargas por TRIAC's*


Módulo Automation
Assinalar sub-fóruns como lidos
FÓRUM
TÓPICOS
MENSAGENS
ÚLTIMA MENSAGEM
Aula 1 - Ethernet Shield
Aula inaugural do Curso Arduino Automation, sobre Ethernet Shield. Verifique nesse tópico sobre o download das bibliotecas EthernetSup. Área reservada para discussões sobre o assunto e melhorias das bibliotecas.
96
392
por ALENCARCIRO
Qui Nov 20, 2014 6:17 pm
Aula 2 - Acionamento de Cargas por Relês
Nesta aula aprendemos sobre relês, conhecemos as formas de conexão com as cargas e pudemos ver em funcionamento a biblioteca EthernetSup que traz botões grandes, que facilitam a utilização com os dedos, em smartphones etc. Aqui poderemos discutir sobre esta aula e os detalhes técnicos das bibliotecas EthernetSup.h e Enc28CoreLib.h
40
153
por franklin lgs
Seg Out 27, 2014 12:05 am
Aula 3 - Controle de Cargas por TRIAC's
Nesta aula veremos como controlar a velocidade do ventilador de teto, através de um Dimmer Digital AC, utilizando um TRIAC para acionamento proporcional da carga. Conheceremos também a Revisão 2 do Módulo Dimmer Digital AC 600V.
29
109
por le_janei
Qui Out 09, 2014 8:04 pm
Aula 4 - Plataforma Supervisório
Nesta aula apresentaremos a Plataforma Supervisório, que é a base do Kit Curso Arduino Automation! Veremos como montar a caixa do Quartinho!
18
70
por franklin lgs
Qua Nov 05, 2014 9:18 pm

Passo 4 – Agora é só localizar o *Tópico do Código*. Repare que vale a pena acompanhar esse tópico para obter novas versões desse código-fonte, que controla os *Módulos Dimmer AC*. O código-fonte não foi publicado neste manual justamente para não ficar desatualizado rapidamente, já que muitas melhorias vem sendo feitas nos códigos! Acompanhe!

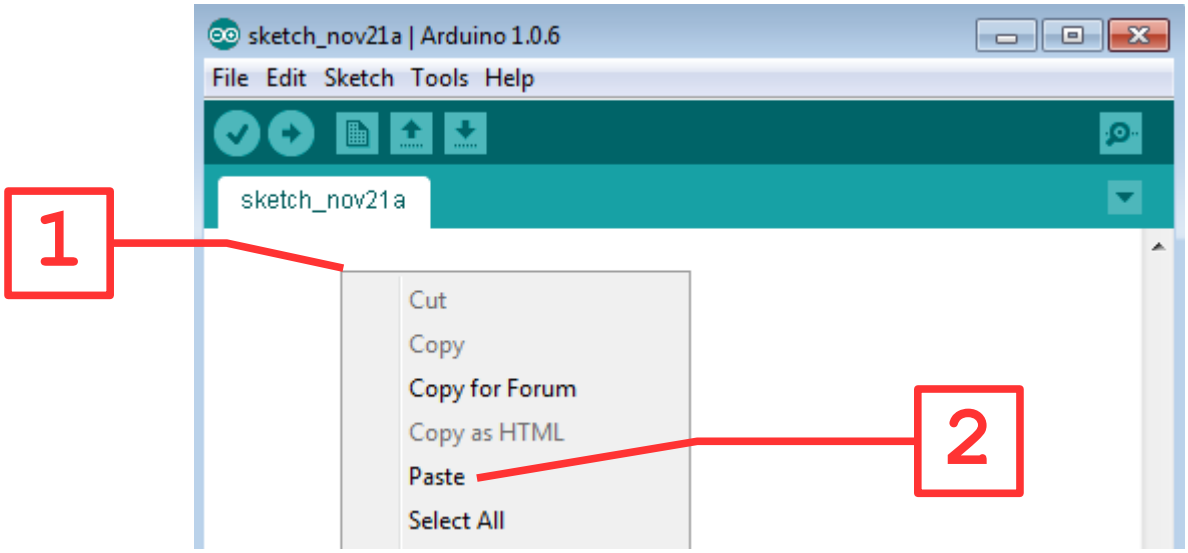
Passo 5 – Na página que abrir, role até o final, pois precisaremos do último código-fonte. Vejamos as etapas para copiá-lo.


Código 4: Serial Dimmer AC 2 - Regula dois Dimmer's AC, ao receber 2 bytes pela serial.
CÓDIGO: SELECIONAR TODOS
1
static volatile unsigned long momentoDisparo = 0;
static volatile unsigned int gatilhoDisparo = 4000;
static volatile unsigned long momentoDisparo2 = 0;
static volatile unsigned int gatilhoDisparo2 = 4000;
static volatile unsigned int potVal = 0;
static volatile unsigned int potVal2 = 0;
const unsigned char portaDimmerAC1 = 5;
const unsigned char portaDimmerAC2 = 6;

Código 4: Serial Dimmer AC 2 - Regula dois Dimmer's AC, ao receber 2 bytes pela serial.
CÓDIGO: SELECIONAR TODOS
3
2
static volatile unsigned long momentoDisparo = 0;
static volatile unsigned int gatilhoDisparo = 4000;
static volatile unsigned long momentoDisparo2 = 0;
static volatile unsigned int gatilhoDisparo2 = 4000;
static volatile unsigned int potVal = 0;
static volatile unsigned int potVal2 = 0;
const unsigned char portaDimmerAC1 = 5;
const unsigned char portaDimmerAC2 = 6;
Copy
Search Google for 'static
Print...
Inspect element

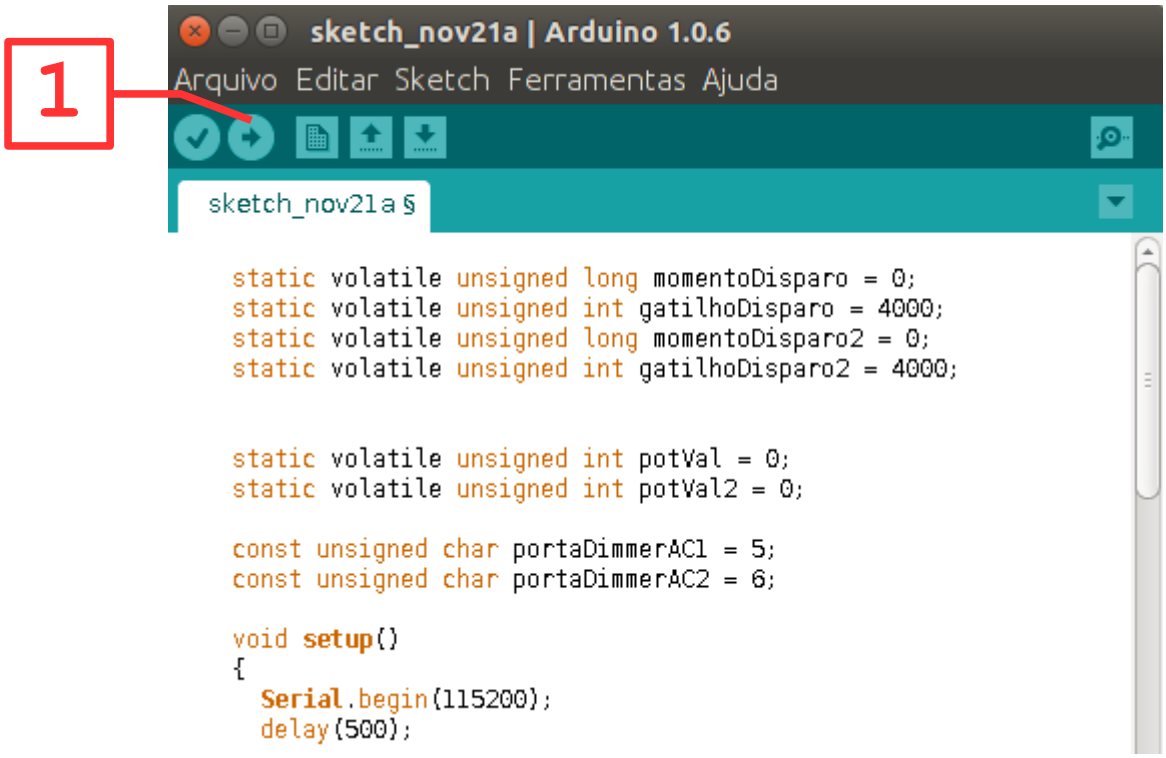
1. Clique no link SELECIONAR TODOS, conforme indicado na imagem
2. Isso fará todo o código-fonte ficar selecionado, escolha um lugar selecionado para clicar com o botão direito do mouse.
3. No menu de contexto que aparecer, escolha a opção “Copy” (Copiar).

Passo 6 – Basta agora abrir o *Arduino IDE* e colar o código-fonte que acabamos de "Copiar" do Fórum da Serial Link.



1. Clique com o botão direito do mouse em uma parte vazia da área de código-fonte.
2. No menu de contexto que aparecer, clique na opção "*Paste*" (Colar).

Passo 7 – Agora basta carregar o código para o ***Arduino Supervisório #2***.


sketch_nov21a | Arduino 1.0.6
Arquivo Editar Sketch Ferramentas Ajuda
1
sketch_nov21a §
static volatile unsigned long momentoDisparo = 0;
static volatile unsigned int gatilhoDisparo = 4000;
static volatile unsigned long momentoDisparo2 = 0;
static volatile unsigned int gatilhoDisparo2 = 4000;
static volatile unsigned int potVal = 0;
static volatile unsigned int potVal2 = 0;
const unsigned char portaDimmerAC1 = 5;
const unsigned char portaDimmerAC2 = 6;
void setup()
{
Serial.begin(115200);
delay(500);

1. Clique no Botão Carregar, indicado na imagem para terminar o processo. Vamos ver as configurações dos *Módulos Dimmer AC* na parte de *Utilização do Kit Automation*, mais adiante, ainda neste manual.

Utilização do Kit Automation

A utilização do ***Kit Curso Arduino Automation*** é bem simples, bastando apenas seguir algumas regras de configuração. Depois que o Kit estiver configurado, normalmente não mexemos mais a não ser que haja uma manutenção ou nova adição a ser feita.

O ***Kit Curso Arduino Automation*** é um Kit Educacional e visa instruir o aluno/instalador enquanto efetua as montagens e configurações. Ou seja: aprenda enquanto monta!

Futuramente lançaremos *Kit's Profissionais*, mas por serem mais avançados, necessitarão de um conhecimento prévio maior.

Este Kit é o lugar certo para começar, inclusive para quem já tem experiências com a *Plataforma Arduino*, mas nunca trabalhou com o chip *ATMega*, fora da *Placa do Arduino*. No ***Kit Curso Arduino Automation***, os *Arduinos Supervisórios* utilizados são baseados na *Plataforma Arduino*, mas possuem *Hardware* próprio da *Serial Link*.

É mais ou menos como aprender a andar de bicicleta, primeiro adicionamos rodinhas para ajudar no equilíbrio. Quando a criança pega o jeito de andar de bicicleta com as rodinhas auxiliares, tiramos as rodinhas; e a criança percebe que sabe andar de bicicleta!

O mesmo acontece com a *Plataforma Arduino*, primeiro o aluno aprende a trabalhar com a *Placa do Arduino*, com auxílio do *Conversor USB/Serial* embutido. Com o passar do tempo, o aluno sente a necessidade de utilizar apenas o chip *ATMega328p*, fora da *Placa do Arduino*. Nesse momento é que as dificuldades começam a aparecer e que o desenvolvedor deve aprender a “andar sozinho”.

A partir de agora veremos algumas opções, que apesar de serem básicas, podem ser um pouco intimidantes para os iniciantes. Vamos lá!

Configurações Básicas

A configuração mais importante do ***Kit Curso Arduino Automation*** é justamente o *Endereço IP*.

O Endereço IP é um tipo de CEP (Zip Code), que identifica um local na Rede de Computadores. Normalmente esse local é representado por um computador. Mas atualmente pode ser uma variedade quase infinita de coisas, por exemplo:

- Computadores
- Roteadores
- Pontos de Acesso
- Arduino's
- Raspberry's Pi
- Console de Jogo (Xbox, PlayStation, Nintendo, etc)
- Set Top Boxes
- etc, etc

Em uma residência tradicional brasileira, temos um Modem da empresa que fornece Internet ligado em um Roteador Wireless. Isso significa que normalmente temos vários aparelhos Wireless passando pelo Roteador e acessando a Internet, ou se comunicando entre si.

Normalmente o Roteador possui o *Endereço IP* inicial da *Rede Local de Computadores*. Esse que nos interessa! Ele pode ser parecido com um dos exemplos abaixo:

1. 192.168.0.1 (o mais comum)
2. 192.168.1.1 (segundo mais comum, mas há controvérsias entre os dois, quem ganha)
3. 10.0.0.1 (roteadores em sub-redes podem apresentar esse tipo de endereço)

Acredito que 99,9% dos casos estão entre os dois primeiros. Para ter certeza disso, utilize um computador conectado ao Roteador, seja por cabo ou por Wireless e digite os comandos abaixo em uma janela de DOS/Bash:

No Windows:

```
> ipconfig
> ping 192.168.0.1
```

No MAC OSX / Linux:

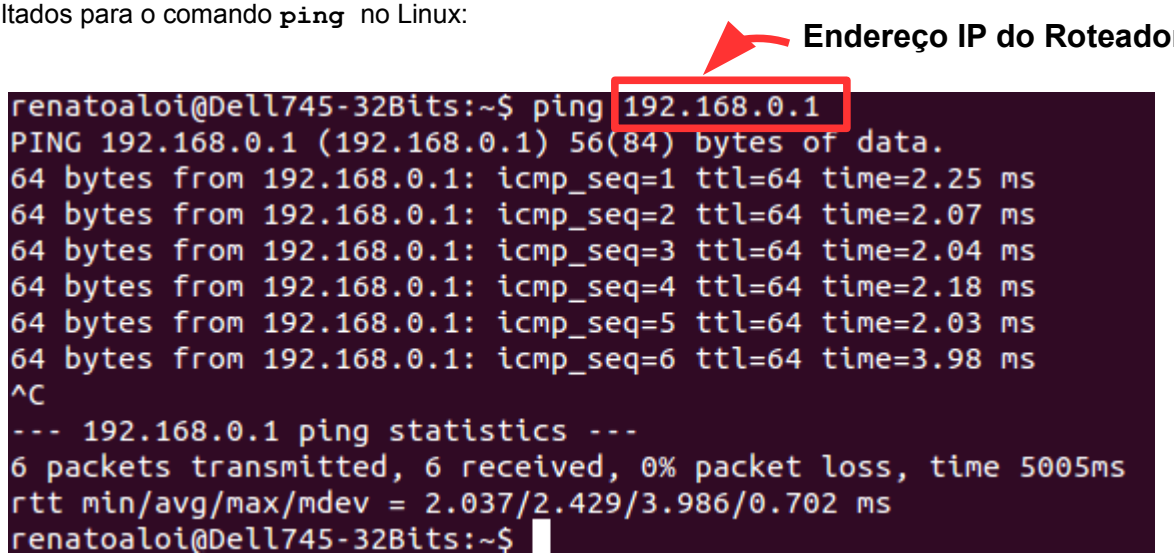
```
$ ifconfig eth0
$ ping 192.168.0.1
```

Meu Endereço IP da Rede Local

Resultados para o comando **`ifconfig`** no Linux:

```
renatoaloi@Dell745-32Bits:~$ ifconfig eth0
eth0      Link encap:Ethernet  Endereço de HW 00:18:8b:e0:96:d1
          inet end.: 192.168.0.151  Bcast:192.168.0.255  Masc:255.255.255.0
          endereço inet6: fe80::218:8bff:fee0:96d1/64 Escopo:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Métrica:1
          pacotes RX:296715 erros:0 descartados:0 excesso:0 quadro:0
          Pacotes TX:191544 erros:0 descartados:0 excesso:0 portadora:0
          colisões:0 txqueuelen:1000
          RX bytes:399097142 (399.0 MB) TX bytes:37362629 (37.3 MB)
          IRQ:16
```

Resultados para o comando **`ping`** no Linux:


Endereço IP do Roteador
renatoaloi@Dell745-32Bits:~$ ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=2.25 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=2.07 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=2.04 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=64 time=2.18 ms
64 bytes from 192.168.0.1: icmp_seq=5 ttl=64 time=2.03 ms
64 bytes from 192.168.0.1: icmp_seq=6 ttl=64 time=3.98 ms
^C
--- 192.168.0.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5005ms
rtt min/avg/max/mdev = 2.037/2.429/3.986/0.702 ms
renatoaloi@Dell745-32Bits:~$

Vejamos:

- Se o Endereço IP do seu computador possui o terceiro byte zero (0)
  - significa que o Endereço IP do Roteador é 192.168.0.1
  - endereços ideais para seu Arduino vão de 192.168.0.20 a 192.168.0.200
- Se o Endereço IP do seu computador possui o terceiro byte zero (1)
  - significa que o Endereço IP do Roteador é 192.168.1.1
  - endereços ideais para seu Arduino vão de 192.168.1.20 a 192.168.1.200

Evite usar endereços abaixos de 20 e tbém acima de 200. O máximo para esse valor do quarto byte é 254, porque 255 é o valor da máscara, enfim...

Este é o mais longe que explicarei sobre Endereços IP neste manual. Pesquise sobre o assunto se você possui dúvidas, pois é muito importante entender essa parte para configurar o Roteador, depois para ligar o Arduino na Internet!

Atenção – Vide este *Tópico do Fórum da Serial Link* para maiores informações sobre como ligar seu Arduino na Internet; discussão sobre configuração de roteamento de portas de Roteadores, em seus mais diversos modelos, siga o link: http://seriallink.com.br/forum/viewtopic.php?f=90&t=2018

Gerenciando Botões Automation

Um dos grandes trunfos do ***Kit Curso Arduino Automation*** é seu HTML pré formatado, de forma que o instalador não precisa se preocupar com essa parte. Basta configurar os botões que deseja apresentar ao usuário, e programar suas funções.

Vamos ver como os *Botões Automation* funcionam e já teremos base para saber como os outros *Elementos Automation* funcionam.

Botões Automation são *Elementos HTML*, pré configurados. Basta adicioná-los uma vez no *Banco de Dados WebDB Automation* e eles sempre estarão lá! Isso também vale para os outros controles, como *Dimmers*, *Logins* etc.

Mais adiante vamos aprender como reiniciar o *Banco de Dados WebDB Automation*, mas por ora, veremos apenas os comandos individuais e suas configurações.

Primeira coisa que devemos saber é que existem dois tipos de botões e são definidos pelas *Macros de Tipos de Botões Automation*. O nome é complicado, mas significa que temos algumas palavras-chaves que podemos utilizar para definir o tipo de botão.

As palavras-chaves que definem os tipos de botão são:

- **`ONOFF_BUTTON`** – Define um botão com dois estados: ligado/desligado
- **`PULSE_BUTTON`** – Define um botão de “pulsar” (exemplos: fechadura eletrônica, campainha, etc)

Acompanhem este *Tópico do Fórum da Serial Link* para maiores informações sobre as *Macros de Tipos de Botões Automation*, segue o link: http://seriallink.com.br/forum/viewtopic.php?f=107&t=2843

Dito isso, podemos apresentar a função de criação dos *Botões Automation*:

```
automation.addButton(char pin, char *text, char type);
```

Onde:

- **`automation`** – Objeto da classe EthernetAutomation
- **`addButton`** – Nome da função que adiciona *Botões Automation* no *Painel de Controle Web*
- **`pin`** – Porta do Arduino atrelada ao botão
- **`text`** – String com o nome de exibição do botão (até 19 caracteres)
- **`type`** – Uma das duas opções: **`ONOFF_BUTTON`** ou **`PULSE_BUTTON`**

Exemplo de um código de Botão para acionar uma lâmpada:

```
const char portaLampada = 5;

EthernetAutomation automation(80);

automation.addButton(portaLampada, "Lampada", ONOFF_BUTTON);
```

A função acima armazena o *Botão Automation* no *Banco de Dados WebDB Automation* e fica registrado pra sempre, mesmo que o *Kit Automation* seja desligado da alimentação! Para reconfigurar um botão, veja a seção de *Reiniciar o Banco de Dados WebDB*.

Esse código-fonte que vimos acima faz parte da configuração e deve ficar em parte dentro da função **`setup()`** do *Arduino*.

Já o código que verifica se o botão foi acionado, deve ser parecido com este abaixo, e ficar dentro da função **`loop()`** do *Arduino*:

```
// Verifica se algum botao foi pressionado
int lastButton = automation.getLastClickedButton();
int buttonState = automation.getButtonState(lastButton);

if (lastButton == portaLampada)
{
      digitalWrite(portaLampada, buttonState);
}
```

Abaixo podemos ver um código completo com apenas um *Botão Automation* para acionamento de uma Lâmpada, ligada através de um *Módulo de Relê*, na *Porta 5* do *Arduino*:

```
#include <EEPROM.h>
#include <EthernetAutomation.h>
#include <EtherEncLib.h>
#define ACTIVATE_LOGIN 1
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 192, 168, 0, 177 };
const char portaLampada = 5;
EthernetAutomation automation(80);
void setup() {
  Serial.begin(115200);
  delay(200);
  Serial.println(F("Serial OK!"));
  automation.begin(mac, ip);
  delay(200);
  Serial.println(F("Ethernet OK!"));
  if (!automation.isWebDBExists()) {
    // Criando WebDB
    automation.addDB();
    // Criando botoes
    automation.addButton(portaLampada, "Lampada", ONOFF_BUTTON);
    Serial.println(F("DB Add OK!"));
  }
  else {
    Serial.println(F("DB Init OK!"));
  }
}
void loop() {
  if (automation.available(ACTIVATE_LOGIN)) {
    Serial.println(F("ETH avail OK!"));
    // Verifica se algum botao foi pressionado
    int lastButton = automation.getLastClickedButton();
    int buttonState = automation.getButtonState(lastButton);
    if (lastButton == portaLampada) {
      digitalWrite(portaLampada, buttonState);
    }
    automation.close();
  }
}
```

Gerenciando Dimmers Automation

Dimmers Automation diferem um pouco dos *Botões Automation* pelo fato deles lidarem com mais estados do que simplesmente: ligado e desligado. Ou seja, *Dimmers Automation* podem assumir valores intermediários entre ligado e desligado, como por exemplo o efeito chamado “Meia Luz”. Este efeito é muito conhecido por ter se popularizado em quartos de crianças, onde é possível ajustar a intensidade da luz, através de um botão tipo “volume”; girando para um lado, aumenta a intensidade da luz, girando para o outro, diminui.

No ***Kit Curso Arduino Automation*** temos este mesmo efeito, mas através de botões do tipo Dimmer Automation. Imagine que você deitou com seu filho e ele dormiu. Mas a luz está te incomodando e você não quer desligar, apenas suavizar. Se você levantar, seu filho vai acordar. Mas tendo o celular a mão, com alguns cliques você pode ajustar a intensidade da luz.

Temos dois (2) ***Módulos Dimmer AC*** no ***Kit Curso Arduino Automation***. Vamos ver como é a função de cadastramento de *Dimmers Automation* no *Painel de Controle Web:*

```
automation.addDimmer(char pin, char *text, char valorIni, char direction, char step);
```

Onde:

- **`automation`** – Objeto da classe EthernetAutomation
- **`addDimmer`** – Função que adiciona *Dimmers Automation* no *Painel de Controle Web*
- **`pin`** – Porta do Arduino a ser acionada
- **`text`** – Texto do dimmer (até 19 caracteres)
- **`valorIni`** – Valor inicial do dimmer
- **`direction`** – Direção inicial do dimmer
- **`step`** – Quantidade de passos por acionamento

Os *Dimmers Automation* possuem uma função adicional para disparar o comando (que é enviado pela Serial), conforme segue:

```
sendDimmerCommand();
```

Repare que o comando não recebe argumentos e não retorna nada. Ele precisa ser executado no início e toda vez que formos atualizar o valor de algum dos *Módulos Dimmer AC*.

Exemplo de um código de Dimmer para regular intensidade de uma lâmpada:

```
const char portaDimmer1 = 98; // utilize numeros perto de 100
static char dimmerVal1 = 127; // vai de 0 a 255

EthernetAutomation automation(80);

automation.addDimmer(portaDimmer1, "Vel Vent: ", 127, -1, 25);
sendDimmerCommand();
```

Vamos fazer uma revisão no código acima.

Na primeira linha já temos um valor estranho para a porta do Arduino.

```
const char portaDimmer1 = 98;
```

Não existe uma porta 98 no Arduino! Sim, mas é um valor virtual de porta, só para atrelar uma coisa na outra. Utilize valores próximos de 100, mas menores que 127.

Ok, na segunda linha, temos uma variável chamada x para guardar o valor da intensidade do Dimmer e ela é iniciada com o valor 127, colocando a intensidade em uma posição mediana.

```
static char dimmerVal1 = 127;
```

Esse valor pode variar entre 0 e 255, inclusive.

Já na quarta linha temos a chamada da função em si! Vamos dar uma olhada nos parâmetros para explicar como funcionam:

```
automation.addDimmer(portaDimmer1, "Vel Vent: ", 127, -1, 25);
```

Sendo:

- **`portaDimmer1`** – valor da porta virtual do Dimmer
- **`"Vel Vent: "`** – texto do Dimmer, para apresentação
- **`127`** – que é o valor inicial da intensidade do Dimmer
- **`-1`** – que é o estado inicial da direção, ou seja, nulo
- **`25`** – é o passo de incremento.

O mais interessante parâmetro dessa função é o **`step`**, que nesse caso foi configurado como 25. Vamos entender...

25 é uma décima parte de (mais ou menos) 255, que é um byte. Então se pensarmos em termos de porcentagem, seria 10% de um byte, por incremento. Fica meio complicado, pois não é base 10.

Mas para resumir, o que interessa é que teremos que apertar umas 10 vezes o botão aumentar (+) ao máximo; ou 10 vezes para diminuir (-) a intensidade até o mínimo. Aumentando esse número de step, diminuímos a quantidade de vezes que teremos que apertar os botões para atingir o máximo ou o mínimo da escala.

Quem tiver dúvidas nessa parte, estudar Lógica Booleana e Aritmética Binária. Ajuda muito conhecer essas matérias. Para facilitar, vamos ver uma tabela de valores que podem ser utilizados no parâmetro **`step`**:

Valor	Correspondente Base 10	Descrição
25	10 %	10 toques *
50	20 %	5 toques *
75	30%	3 toques *
125	50%	2 toques *

* para ir de um extremo no outro na faixa de intensidade entre 0 ~ 255.

E finalmente, ainda na análise do código, temos a chamada da função **`sendDimmerCommand()`** que envia efetivamente o comando (que na verdade é o valor da variável **`dimmerVal1`**) para o *Arduino Supervisório #2*, responsável pelo controle dos *Módulos Dimmer AC*.

```
sendDimmerCommand();
```

Repare que essa função não tem o objeto **`automation`** na frente, pois não é um método da classe *EthernetAutomation*, mas sim uma função auxiliar, conforme segue:

```
void sendDimmerCommand()
{
  Serial.write(dimmerVal1);
}
```

Esta função faz parte do código-fonte exemplo e simplesmente envia o valor da variável x para a Serial. O outro *Arduino Supervisório #2* que é "responsável por entender" o comando e ajustar a posição dos *Dimmers* na escala de intensidade.

Vamos ver então o exemplo completo do *Dimmer Automation* no *Painel de Controle Web*:

```
// Verifica se algum botao foi pressionado
int lastButton = automation.getLastClickedButton();
int buttonState = automation.getButtonState(lastButton);

if (lastButton == portaLampada)
{
      digitalWrite(portaLampada, buttonState);
}
```

Abaixo podemos ver um código completo com apenas um *Botão Automation* para acionamento de uma Lâmpada, ligada através de um *Módulo de Relê*, na *Porta 5* do *Arduino*:

```
#include <EEPROM.h>
#include <EthernetAutomation.h>
#include <EtherEncLib.h>
#define ACTIVATE_LOGIN 1
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
byte ip[] = { 192, 168, 0, 177 };

const char portaDimmer1 = 98;
static char dimmerVal1 = 127;

EthernetAutomation automation(80);
void setup() {
  Serial.begin(115200);
  delay(200);
  Serial.println(F("Serial OK!"));
  automation.begin(mac, ip);
  delay(200);
  Serial.println(F("Ethernet OK!"));
  if (!automation.isWebDBExists()) {
    // Criando WebDB
    automation.addDB();
    // Criando dimmers
    automation.addDimmer(portaDimmer1, "Vel Vent: ", 127, -1, 25);
    Serial.println(F("DB Add OK!"));
  }
  else {
    Serial.println(F("DB Init OK!"));
  }
  sendDimmerCommand();
}


void loop() {
  if (automation.available(ACTIVATE_LOGIN)) {
    Serial.println(F("ETH avail OK!"));
    // Verifica se algum botao foi pressionado
    int lastButton = automation.getLastClickedButton();
    if (lastButton == portaDimmer1) {
      dimmerVal1 = sup.getDimmerValue();
      sendDimmerCommand();
    }
    automation.close();
  }
}

void sendDimmerCommand()
{
  Serial.write(dimmerVal1);
}
```

Gerenciando Sensores Automation

Não disponível ainda nessa revisão do manual. Em desenvolvimento.

Gerenciando Logins Automation

O sistema de Logins Automation é uma nova ferramenta implementada bem recentemente, por solicitação antiga dos membros da comunidade do Curso Arduino.

Toda implementação dos Logins foi feita nos moldes dos Botões e dos Dimmers, sendo assim, podemos adicionar *Logins Automation* no *Banco de Dados WebDB Automation*.

Vimos na parte de instalações o *Formulário de Login*, conforme imagem abaixo:

Nesta tela podemos usar o Login Administrativo ou algum Login cadastrado pela função que veremos a seguir.

O Login Administrativo possui os seguintes dados fixos:

login: **`admin`**
senha: **`1234`**

Mas é importante saber que a tela, da imagem acima, só aparece se ativarmos o *Sistema de Logins Automation*. Para isso existe uma Macro especial chamada **`ACTIVATE_LOGIN`** que controla o funcionamento (ou não) dos *Logins*.

Podemos ver essa diferença ao compararmos os códigos-fontes **`Automation.ino`** e **`AutomationLogin.ino`**, que acompanham a biblioteca *EthernetAutomation*.

Vejamos o código-fonte sem *Sistema de Logins Automation*:

```
void loop() {
  if (automation.available()) {
    automation.close();
  }
}
```

Agora o código-fonte com *Sistema de Logins Automation*:

```
#define ACTIVATE_LOGIN 1

void loop() {
  if (automation.available(ACTIVATE_LOGIN)) {
    automation.close();
  }
}
```

Essa simples diferença entre os códigos é a chave para ativar o sistema de Logins Automation:

- No primeiro código, ao acessarmos o *Endereço IP* do *Kit Automation*, seríamos apresentados diretamente ao *Painel de Controle Web*;
- Já no segundo código, antes de mais nada, seríamos apresentados ao *Formulário de Login*. Só ao digitar algum login válido poderíamos acessar o *Painel de Controle Web*.

Vamos ver então a função que adiciona Logins Automation ao Painel de Controle Web.

```
automation.addLogin(char* _login, char* _pass, ulong timeout);
```

Onde:

- **`automation`** – Objeto da classe EthernetAutomation
- **`addLogin`** – função que adiciona os Logins no Banco de Dados WebDB
- **`_login`** – Login a ser adicionado (máximo de 9 caracteres)
- **`_pass`** – Senha do login (máximo de 9 caracteres)
- **`timeout`** – Tempo de duração da sessão de Login. Em minutos.

Nada mais precisa ser feito. Depois de configuradas as opções mostradas aqui, o *Sistema de Logins Automation* cuida do resto.

Gerenciando Banco de Dados WebDB Automation

Uma das modificações da nova biblioteca *EthernetAutomation*, desenvolvida especialmente para o lançamento do ***Kit Curso Arduino Automation***, é a adição de um *Banco de Dados* chamado *WebDB*.

Esta nova ferramenta nos permite gravar de forma "mais permanente" nossos *Botões*, *Logins*, *Dimmers*, e futuramente, *Sensores*!

Isso sem falar no fato que o novo *WebDB* é capaz de armazenar muito mais informação que tínhamos disponíveis na memória RAM. Agora podemos gravar quase 30 Botões/Dimmers/etc; e 8 Logins!

Outro motivo para adoção desse novo *Banco de Dados WebDB* é o fato de que as informações não se perdem quando o *Kit Automation* sofre uma perda de energia! Quando voltar, o sistema estará com todas as informações prontas! Ou seja, os dados do *WebDB* não são perdidos quando o *Kit Automation* é desligado!

Pelo motivo do *WebDB* registrar as informações de forma permanente, precisamos utilizar alguns recursos para que as informações não se repitam; ou quando queremos limpar e gravar novamente as informações do ***Banco de Dados WebDB Automation***.

Primeiro vamos ver a função que verifica se o *Banco de Dados WebDB* existe. O sistema do *Kit Automation* utiliza dois bytes de controle para reconhecer o formato do seu *Banco de Dados*. Veremos como utilizar esses bytes para reiniciar o *WebDB*.

Função para verificar se o *Banco de Dados WebDB* já está instalado no *Kit Automation*:

```
automation.isWebDBExists()
```

Onde:

- **`automation`** – Objeto da classe *EthernetAutomation*
- **`isWebDBExists`** – Nome da função que verifica se o *Banco de Dados WebDB* existe

Essa função é utilizada normalmente na forma negativa e dentro de um **`if`**. Vejamos um exemplo:

```
EthernetAutomation automation(80);

if (!automation.isWebDBExists()) {
  Serial.println(F("DB Nao Existe!"));
}
else {
  Serial.println(F("DB Existe OK!"));
}
```

Podemos ver pelo código-fonte acima que o funcionamento é bem simples, retornando verdadeiro se o Banco de Dados WebDB existe e falso, caso contrário.

Essa função é muito importante para impedir que os *Botões*, *Logins* e *Dimmers* sejam cadastrados mais de uma vez. Por isso utilizamos a função de criação do banco de dados dentro de um **`if`**, conforme vimos acima.

Função de criação do *Banco de Dados WebDB*:

```
automation.addDB();
```

Onde:

- **automation** – Objeto da classe *EthernetAutomation*
- **addDB** – Nome da função que cria o *Banco de Dados WebDB*

Atenção – É muito importante que essa função de criação do *Banco de Dados WebDB* fique dentro de um **if** que verifique se o *WebDB* já existe! Senão toda vez que o *Arduino* reiniciar, o *Banco de Dados* será recriado, limitando a vida útil da memória *EEPROM*, onde os dados são armazenados.

Exemplo de utilização da função de criação do *Banco de Dados WebDB*:

```
EthernetAutomation automation(80);

if (!automation.isWebDBExists()) {
  // criando banco apenas uma vez
  automation.addDB();
}
```

Repare que a função **addDB()** fica dentro do **if** que verifica se o *Banco de Dados WebDB* existe. Isso vale também para todos os outros elementos estudados até agora! Todas funções **addLogin()**, **addButton()**, **addDimmer()** etc, devem ficar dentro desse **if**!

Vejamos um exemplo completo com a criação de cada um dos elementos:

```
EthernetAutomation automation(80);

if (!automation.isWebDBExists()) {
  // criando banco apenas uma vez
  automation.addDB();
  automation.addLogin("seunome", "seuasenha", 5);
  automation.addButton(portaLampada, "Lampada", ONOFF_BUTTON);
  automation.addDimmer(portaDimmer2, "Int Lamp: ", 127, -1, 25);
}
```

Trabalhando dessa forma, não duplicamos os registros do *Banco de Dados WebDB* e também não precisamos ficar alimentando os dados toda vez que o *Arduino* reinicia!

Em contra-partida, precisamos utilizar uma técnica para reiniciar o *WebDB*, já que reiniciar o *Arduino* simplesmente não vai apagar os dados.

Para isso, foram criadas duas *Macros*, representando dois bytes de controle. Podemos encontrar as definições dessas Macros no arquivo: **`WebDB.h`**, dentro da biblioteca *EthernetAutomation*. Vejamos como é esse código-fonte:

```
#define CONTROL_WORD_H 0xC0
#define CONTROL_WORD_L 0xA7
```

Repare que é bem simples, apenas define duas palavras-chaves que o sistema de criação do *Banco de Dados WebDB* utiliza para saber se os dados são válidos.

Portanto basta alterar qualquer um dos valores para invalidar o *Banco de Dados WebDB*, por exemplo, se mudarmos o valor **`CONTROL_WORD_L`** para **`0xA8`** (lembre-se que são valores hexadecimais), os dados serão invalidados e o *WebDB* será recriado na próxima vez que o *Arduino Supervisório* reiniciar!

```
#define CONTROL_WORD_H 0xC0
#define CONTROL_WORD_L 0xA8
```

Atenção 1 – Lembre-se disso quando alterar um texto de um Botão Automation ou criar um Login Automation novo! Você precisa invalidar o WebDB para que as alterações sejam aplicadas!

Atenção 2 – O Banco de Dados WebDB guarda seus dados e informações de controle na EEPROM, que é lenta em termos computacionais. Então a criação do WebDB demora cerca de 30 segundos! Não interrompa esse processo reiniciando a Serial para não prejudicar a criação dos dados, quando o Arduino reiniciar! Essa operação só acontece uma única vez (toda vez que alteramos as Macros de controle).

Problemas Conhecidos

Em desenvolvimento.

Ligações Na Rede Elétrica

Atenção 1 – Se você for menor de idade, não vá adiante neste manual sem supervisão de um adulto! Você pode montar tudo até aqui! Inclusive carregar os códigos-fontes nos Arduino's Supervisório sem precisar ligar o ***Kit Curso Arduino Automation*** na *Rede Elétrica*; pois o ***Arduino FTDi Basic*** já fornece os 5V, na hora de carregar a programação.

Atenção 2 – A *Rede Elétrica* pode matar! Muita atenção nas ligações daqui para frente!

Campanha da AES Eletropaulo Sobre Perigos da Rede Elétrica

Alimentação Principal

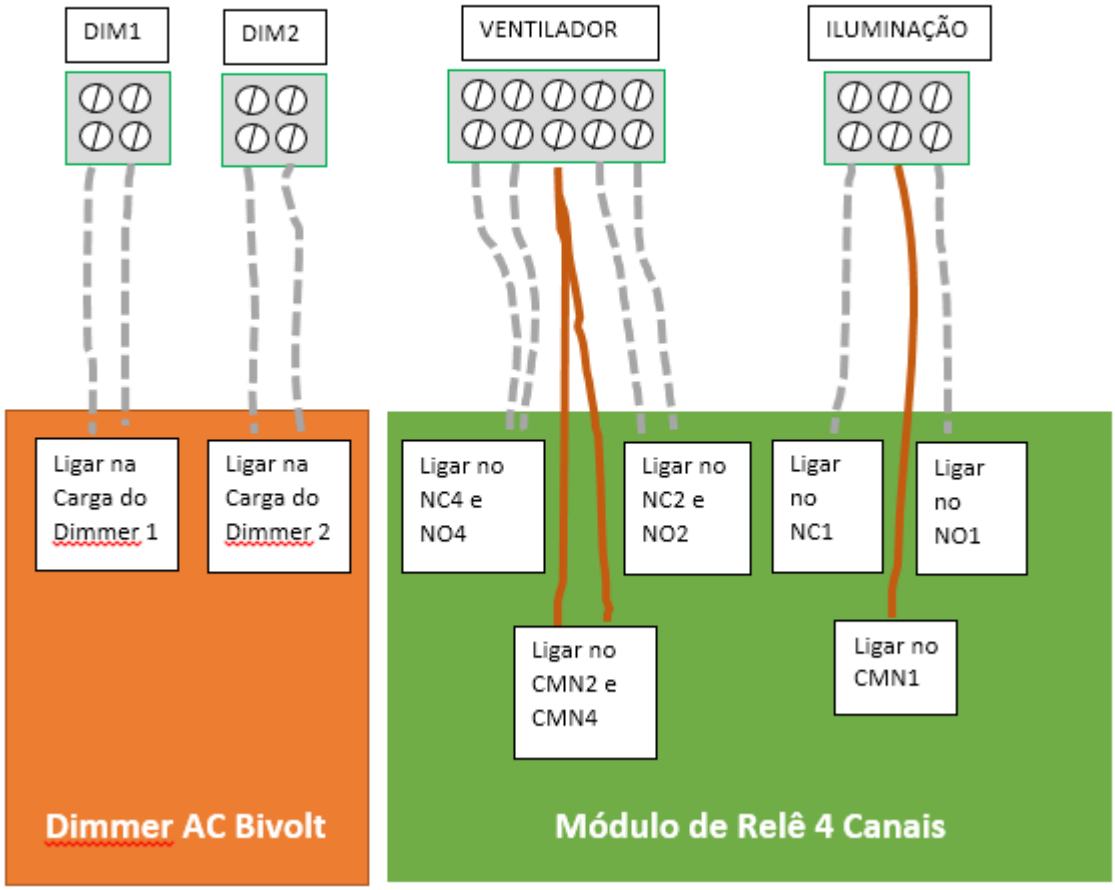
Para a correta montagem do ***Kit Curso Arduino Automation***, vamos precisar de alguns equipamentos e ferramentas adicionais, que não são fornecidas.

- Lista de Opções
- Com SubOpções
  - Assim

Diagramas de Ligações do Chicote Principal

O principal significado do ***Kit Curso Arduino Automation*** é justamente a conexão de seus atuadores com a Rede Elétrica. Vamos ver agora diversas formas de ligações possíveis através do *Chicote Principal Automation*.

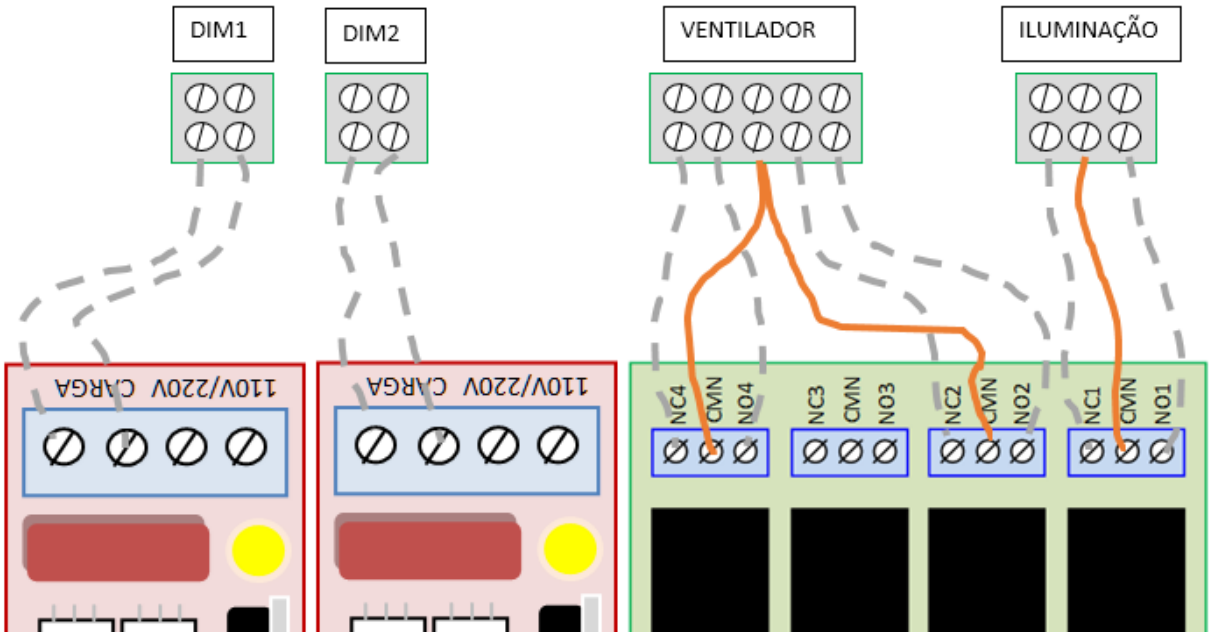
Vamos primeiro lembrar como esse ***Chicote Principal Automation*** é, em sua mais atual revisão:



Repare que temos:

- 2x Conectores de *Dimmer AC*
- 1x Conector de *Ventilador*
- 1x Conector de *Iluminação*

As ligações ficam mais fáceis de serem visualizadas neste outro diagrama em que as placas aparecem:

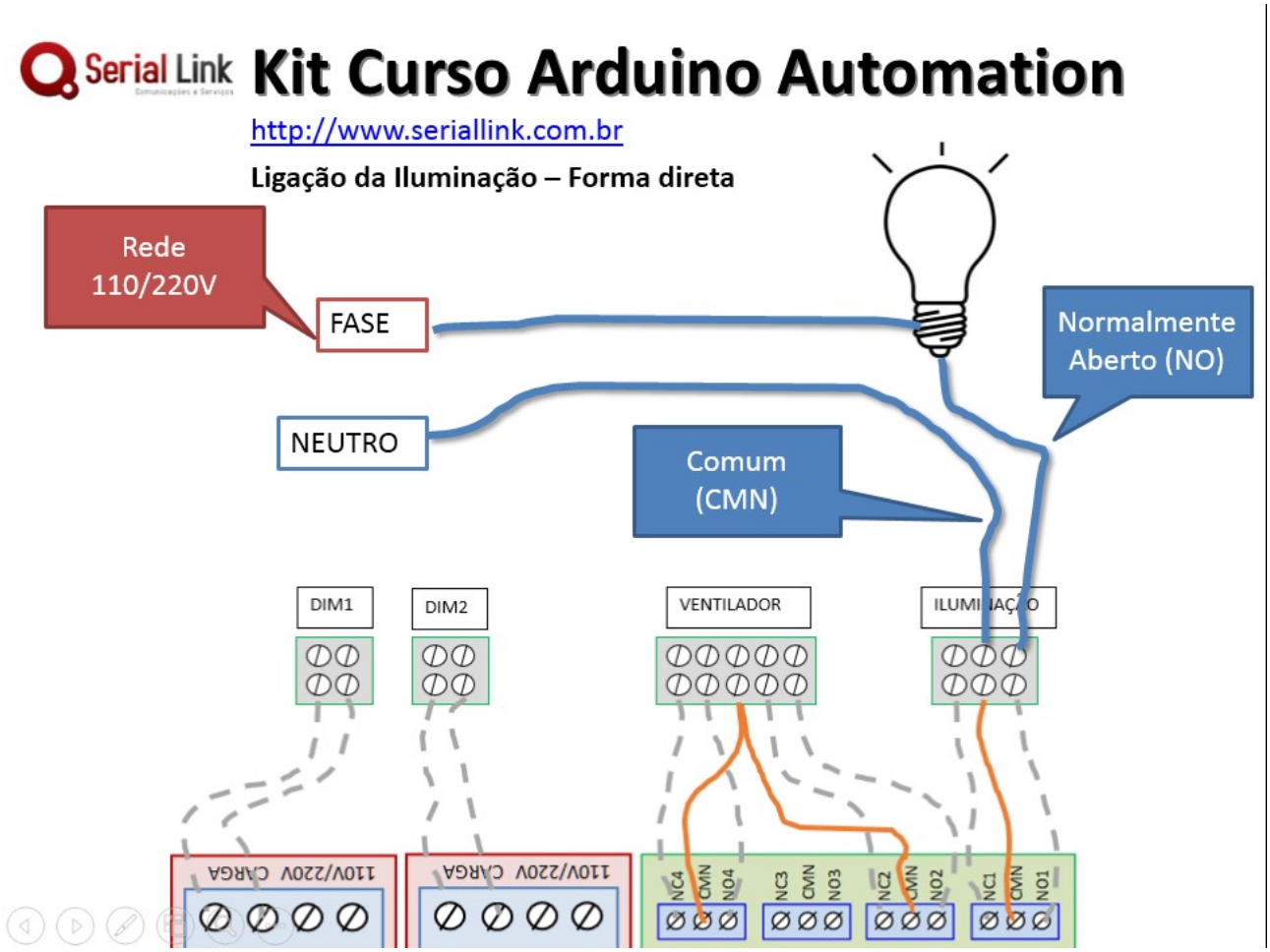


Agora ficou bem mais fácil conectar o Chicote Principal Automation nas placas e também efetuar as conexões com os dispositivos que serão acionados.

Vamos ver logo quais opções temos de ligações.

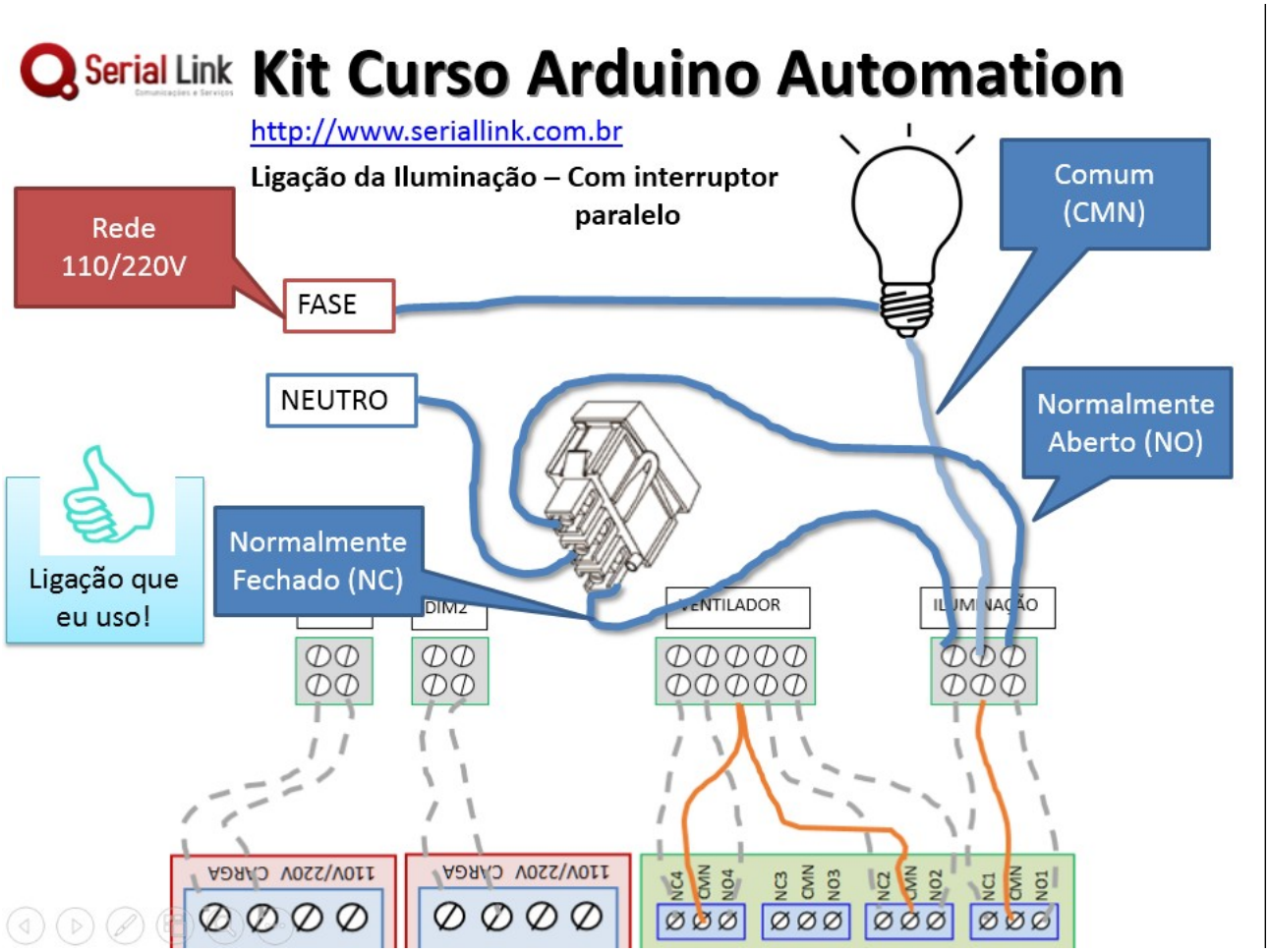
Ligação da Iluminação – Forma Direta

Neste diagrama podemos ver a forma direta de ligação de uma *Lâmpada* (ou dispositivo AC qualquer) no *Canal 1* do ***Módulo de Relês de 4 Canais***.



Ligação da Iluminação – Com Interruptor Em Paralelo

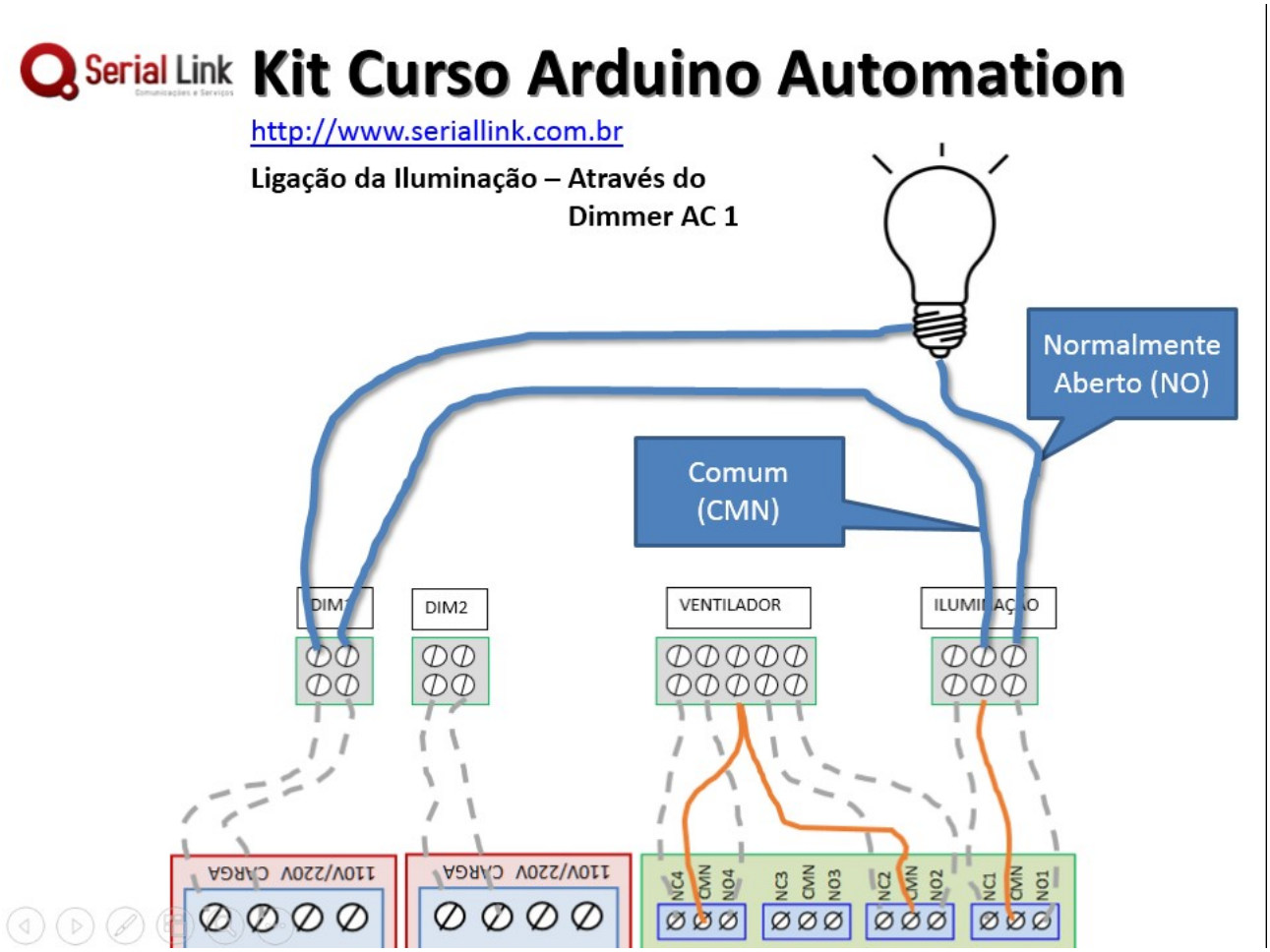
Já neste outro esquema de ligação, podemos perceber que um interruptor, em paralelo com o *Canal 1* do ***Módulo de Relês de 4 Canais***, adiciona uma funcionalidade extra! O usuário pode optar entre o interruptor ou os *Botões Automation* do ***Painel de Controle Web***.



Ligação da Iluminação – Direta Com Controle de Intensidade

Neste modelo, temos a utilização de um dos ***Módulos Dimmer AC Bivolt*** da *Serial Link*, que garantem acionamento proporcional a cargas de filamento e magnéticas (ou seja, lâmpadas incandescentes e motores AC).

Quando utilizamos o ***Módulo Dimmer AC Bivolt #1*** (são dois, no Kit), já contamos com alimentação e não precisamos ligar no *Neutro* ou *Fase* da *Rede Elétrica*. Os próprios *Dimmer's AC* (ambos) já estão conectados na *Rede Elétrica*.



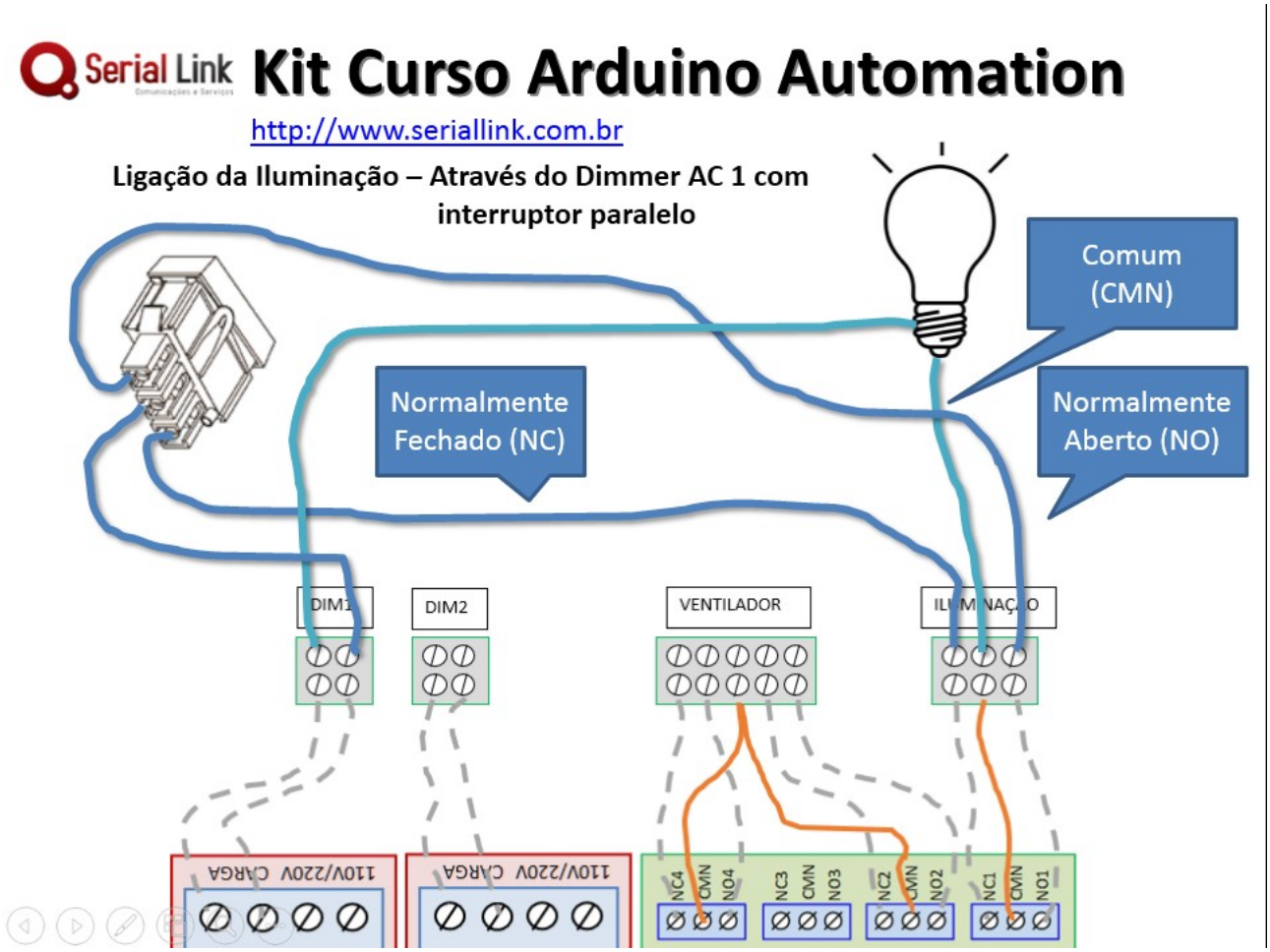
No exemplo acima, poderia ser utilizado o ***Módulo Dimmer AC Bivolt #2***, só que invertendo os controles no ***Painel de Controle Web***.

Ligação da Iluminação – Com Controle De Intensidade E Interruptor Em Paralelo

Esta ligação é um pouco mais complicada, mas o ***Chicote Principal Automation*** facilita tudo!

Perceba que as conexões são bem diretas e não exigem muito malabarismo.

Atenção – Você já deve ter percebido que, nesse ponto da montagem, qualquer conexão mais complicada pode levar horas! Isso pois já temos 99% do Kit montado e tudo fica mais frágil nesse momento! Leve o tempo que precisar para se adaptar e nos envie sugestões de alterações e também suas experiências de montagem! Poste no *Tópico de Projetos dos Membros* do *Fórum da Serial Link*, segue link: http://seriallink.com.br/forum/viewforum.php?f=68

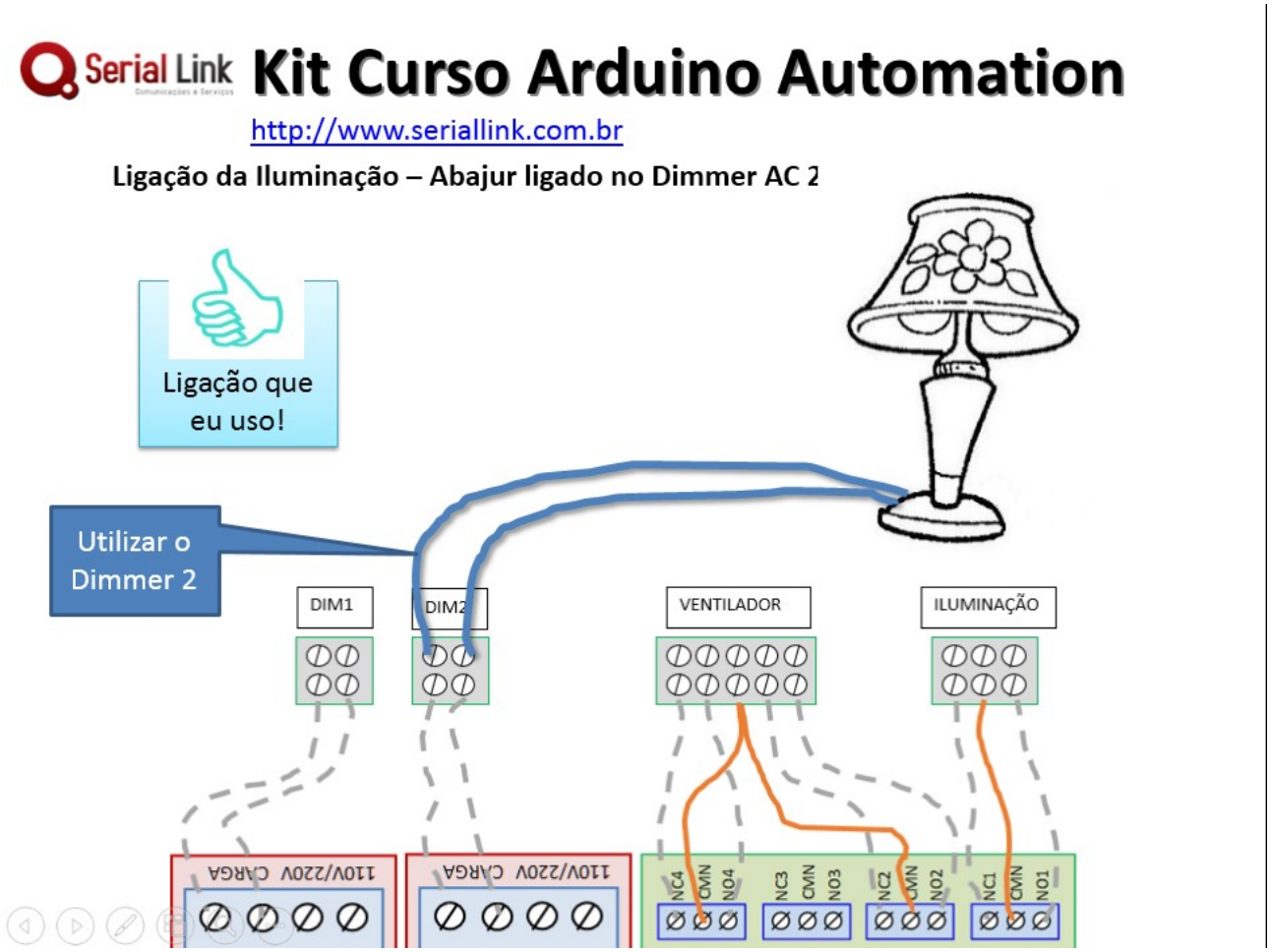


Ligação da Iluminação – Abajur De Leitura

Esta ligação, assim como todas as outras, é uma sugestão para um *Abajur de Leitura*, que pode ter o brilho de sua lâmpada regulado pelo ***Módulo Dimmer AC Bivolt #2***.

Esse tipo de ligação é bem interessante, pois normalmente o abajur possui um *Interruptor Integrado*, mas não possui regulagem de intensidade do brilho!

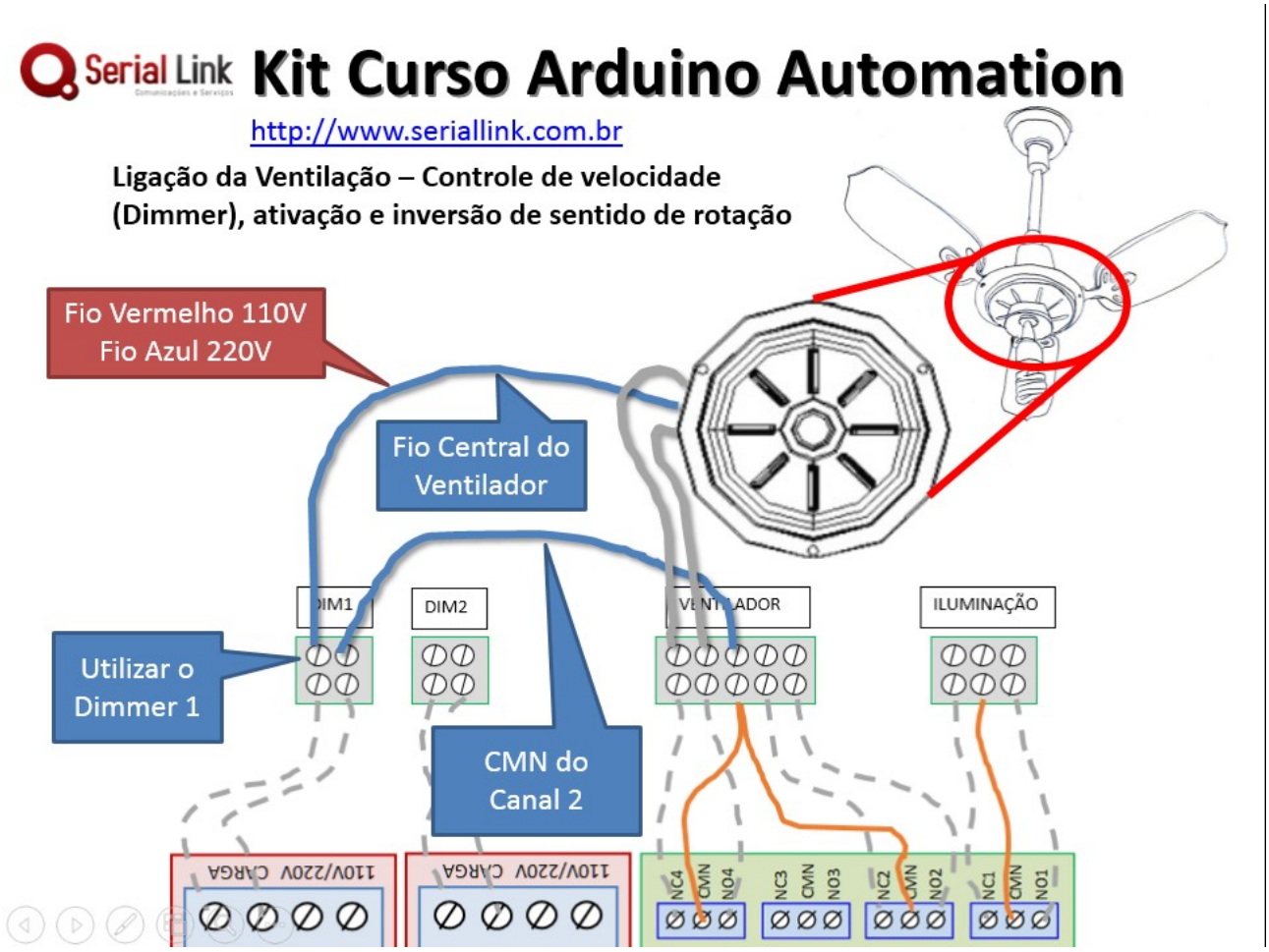
O ***Kit do Curso Arduino Automation*** pode resolver isso! E você nem precisa levantar da cama para regular o abajur, podendo fazer isso através do seu celular!



Ligação Da Ventilação – Controle Da Ativação, Velocidade e Sentido

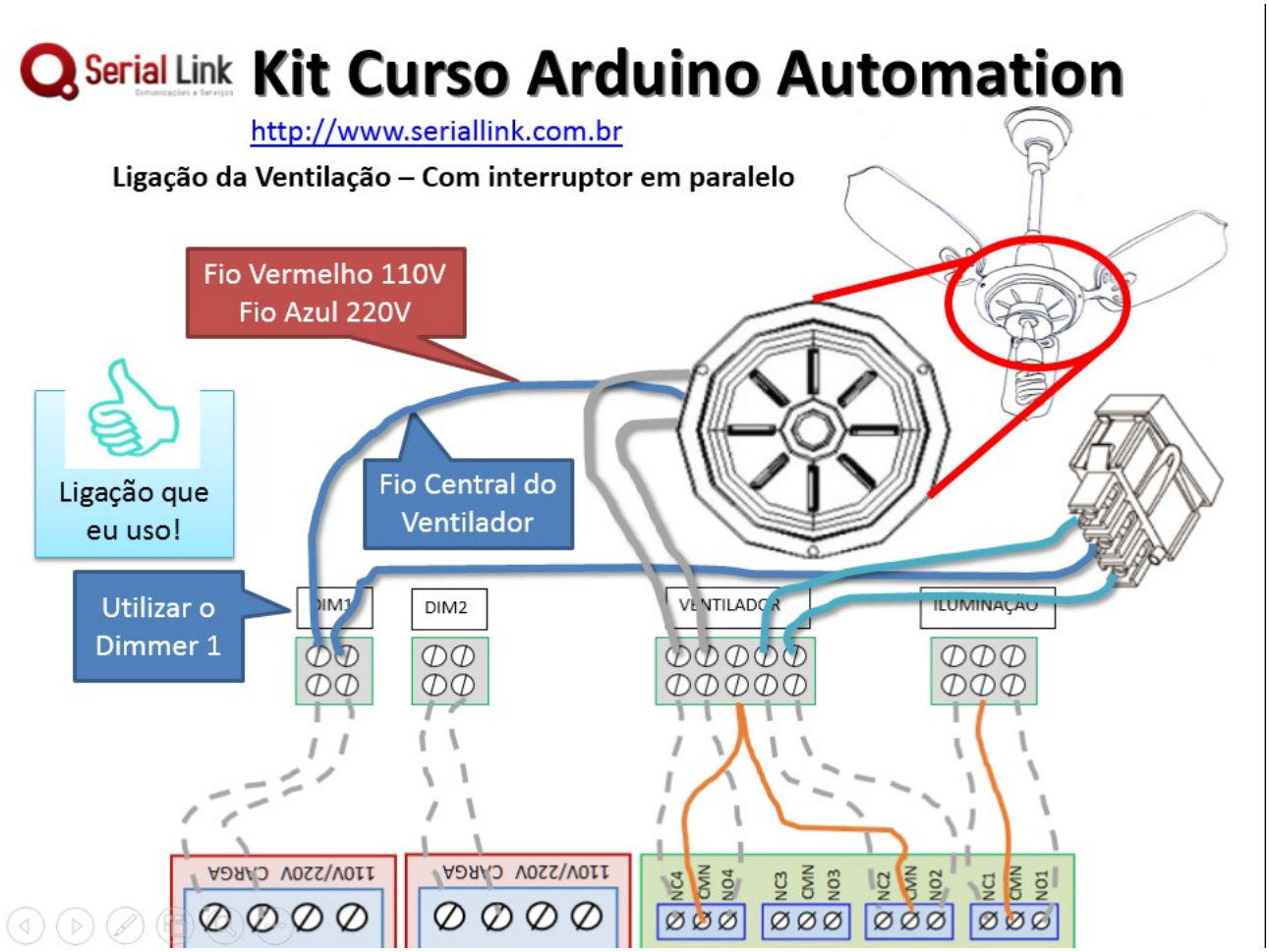
A partir de agora veremos as ligações da parte de ventilação. Esse diagrama considera a ligação de um *Ventilador de Teto* que possui três fios, sendo um central e dois laterais; que determinam o sentido.

Atenção – Não esqueça de ligar o capacitor de partida nos fios laterais!



Ligação Da Ventilação – Com Interruptor Em Paralelo

Nesta ligação, podemos ver o *Ventilado de Teto* conectado em paralelo com um *Interruptor*. Dessa forma o usuário possui duas opções de acionamento do ventilador, através do interruptor, ou pelo ***Painel de Controle Web***.



Sugestão de Passagem da Fiação

Em desenvolvimento.

Anexos

Anexo 1 – Lista de Componentes

Lista Geral

Qtde.	Componente	Descrição
1	Manual Completo Ilustrado	Este documento.
4	DVD do Curso Arduino Automation	
1	Caixa do Sistema VDI da Tigre	30x30cm Sobrepor na parede (Ver lista de conteúdo)
1	Fechadura Eletrônica 12V	Acompanha acessórios de instalação.
2	Caixas de Componentes Serial Link	Ver lista de conteúdo.

Conteúdo da Caixa do Sistema VDI Tigre

Qtde.	Componente	Descrição
1	Corpo do Quadro Sistema VDI	Chamado de “Painel de Fundo” ou simplesmente “Painel”, neste manual.
1	Moldura com Porta	Acompanha 4 parafusos de fixação da moldura.
4	Parafusos de fixação	Fixam a Caixa do Sistema VDI na parede.
1	Cartela de Etiquetas	Sem muita utilidade

Conteúdo da Caixa de Componentes Serial Link #1

Qtde.	Componente	Descrição
1	Chicote Principal Automation	Nova revisão! Muito mais organiizado!
2	Kits Expansores de Porta Automation	Cada Kit contém 2 Adaptadores, sendo um de 2 alojamentos e outro de 3 alojamentos.
1	Chicote de Alimentação Automation + 2 Fontes de 12V / 1A	Completo, contendo todas partes do chicote! Acompanha Tomadas já conectadas, acabamento em organizador retrátil e 2 Fontes de Alimentação 12V / 1A
1	Chicote de Fechadura 12V	Completo, contendo todas partes do chicote!
1	Kit de Rabichos Automation	Completo, com todos os Rabichos necessários!
4	Parafusos de Fixação das Tomadas	Parafusos de fixação das Tomadas T1 e T2 do Chicote de Alimentação Automation
2 Metros	Organizador Retrátil	Tipo “spagetti”
1	Cabo de Rede	

Conteúdo da Caixa de Componentes Serial Link #2

Qtde.	Componente	Descrição
1	Arduino FTDi Basic	
2	Arduino Supervisório	
2	Chip ATMega328p	
1	Módulo de Relê 4 Canais	
2	Módulo Dimmer AC Bivolt	
2	Distribuidor de Alimentação Supervisório	
1	Adaptador Pino/Borne Supervisório	
1	Módulo Ethernet Enc28J60	
25	Cintas Plásticas	
1	Cabo USB	